

Travail de Bachelor 2017

SAP Core Data Services (CDS)



Étudiante : Marlène Duc

Professeur : Dr. Werner Maier

Déposé le : 31.07.2017

Résumé

SAP Core Data Services (CDS), apparu avec SAP HANA, est considéré comme la prochaine génération utilisée pour la définition et l'accès aux données SAP. Basée sur SQL, cette nouvelle infrastructure propose tant des améliorations sur les fonctions et expressions existantes que de nouvelles fonctionnalités.

L'objectif de ce travail est de se familiariser avec le concept, de développer un prototype mettant en avant les différentes fonctionnalités du CDS, puis de déterminer s'il est plus avantageux de passer à SAP CDS ou de rester sur *l'ABAP Dictionary*.

Pour atteindre cet objectif, la première partie du travail vise à mettre en avant les différentes technologies utilisées par SAP, à comprendre le concept SAP CDS et à identifier les différentes fonctionnalités disponibles.

Dans un deuxième temps, un prototype ABAP CDS a été mis en place pour mettre en pratique les fonctionnalités décrites dans la partie théorique. Un deuxième prototype, basé sur HANA CDS, a également été mis en place afin d'illustrer la création de tables grâce au CDS.

A l'issue de la présente étude, on peut constater que les avantages de SAP CDS, tels qu'une gestion des autorisations améliorée et l'introduction des annotations, rendent ce concept plus attractif en comparaison avec *l'ABAP Dictionary*.

La gestion du projet s'est basée sur la méthodologie de travail Agile. Toutefois, le travail n'étant pas un projet de groupe, la méthodologie a été adaptée.

Mots-clés : SAP, SAP CDS, ABAP CDS, HANA CDS, Travail de Bachelor,

Avant-propos

Avec l'arrivée de la plate-forme SAP HANA, un changement de paradigme a eu lieu dans la manière de développer les applications métier chez SAP. Le but est d'obtenir de meilleures performances dans l'accès aux bases de données (Swanepoel, New Data Modeling Features in SAP NW ABAP 7.4 SP5, 2017).

Le travail de Bachelor proposé par le Centre de Compétences SAP de la HES-SO Valais/Wallis, supervisé par le Professeur Dr. Werner Maier, consiste à étudier les principes du CDS, les nouvelles fonctionnalités existantes, à proposer un prototype afin de mettre en avant l'ABAP CDS et, enfin, à proposer un prototype HANA CDS.

Le travail proposé constitue un premier pas dans l'étude de ce nouveau concept. Il est avant tout théorique, axé sur l'analyse des spécificités du SAP CDS, mais offre également une dimension pratique, par le développement de prototypes.

Remerciements

Je tiens à remercier le Professeur Dr. Werner Maier d'avoir proposé ce sujet et de m'avoir suivie tout au long de ce travail de Bachelor.

Je tiens également à remercier Monsieur Frédéric Morand, professeur et membre de l'équipe du Centre de Compétences SAP de la HES SO Valais/Wallis, qui a toujours été disponible pour m'aiguiller et répondre à mes questions.

Mes remerciements vont également aux personnes qui m'ont soutenue durant la réalisation de ce travail, mais également durant mon cursus scolaire. Merci à ma mère, Jacqueline Duc, et à Yannick Romailier qui ont consacré du temps à la relecture de ce rapport.

Table des matières

Liste des figures	VI
Liste des abréviations	VIII
1. Introduction	1
2. Technologies	2
2.1. SAP HANA Platform	2
2.2. SAP HANA	3
2.3. S/4HANA	4
2.4. ABAP Dictionary	5
2.5. Open SQL	6
2.6. OData	6
3. SAP CDS	7
3.1. Concept	7
3.2. Prérequis	8
3.3. ABAP CDS	9
3.4. HANA CDS	9
3.5. Différences entre ABAP CDS & HANA CDS	10
4. Fonctionnalités SAP CDS	11
4.1. Jointures	11
4.2. Associations	11
4.3. Annotations	12
4.3.1. Annotations ABAP	12
4.3.2. Annotations spécifiques à des frameworks	13
4.4. Autorisations	14
4.5. Extension de vues	15
5. Prototype ABAP CDS	16
5.1. Configuration de l'environnement Eclipse	16
5.1.1. Outils de développement ABAP	16
5.1.2. Nouveau projet ABAP	19
5.2. Création d'une entité CDS	23
5.3. Utilisation d'une vue CDS dans un programme ABAP	28
5.4. Annotations	28
5.5. Jointures	29
5.6. Associations	30
5.7. Autorisations	31
5.8. Extension de vues	34
5.9. Paramètres	35
5.10. Gestion des langues	37
6. Prototype HANA CDS	38
6.1. Configuration de l'environnement Eclipse	38
6.1.1. Outils de développement HANA	38
6.1.2. Ajouter un système SAP HANA à Eclipse	39

6.2. Création des tables.....	41
7. Avantages et inconvénients par rapport à l'ABAP Dictionary	46
8. Conclusion.....	47
8.1. Synthèse générale.....	47
8.2. Evolutions futures	47
Références	48
Annexe I : Tables de la base de données	52
Annexe II : Objet d'autorisation	54
Annexe III : Aides et astuces de développement	55
Annexe IV : Insertion de données dans les tables HANA	58
Annexe V : Gestion de projet	63
Annexe VI : Cahier des charges.....	64
Annexe VII : Détails des heures	65
Annexe VIII : Rapport hebdomadaire	66
Contacts.....	67
Déclaration sur l'honneur.....	68

Liste des figures

Figure 1: Plate-forme SAP HANA.....	2
Figure 2 : Données en ligne / Données en colonnes	3
Figure 3: SAP vs SAP S4/HANA	4
Figure 4 : Repository SAP	5
Figure 5 : Evolution Open SQL	6
Figure 6 : Core Data Services	7
Figure 7 : Expressions et fonctions CDS.....	8
Figure 8 : Jointures.....	11
Figure 9 : Annotations CDS	13
Figure 10 : Autorisations CDS	14
Figure 11 : Extension de vues CDS.....	15
Figure 12 : Installation ADT Tools - Install New Software.....	16
Figure 13 : Installation ADT Tools - Add link	17
Figure 14 : Installation ADT Tools - Add Repository	17
Figure 15 : Installation ADT Tools - Available Software	18
Figure 16 : Installation ADT Tools - Review Items	18
Figure 17 : Installation ADT Tools - Review Licenses	19
Figure 18 : Configuration Eclipse - Modification perspective	19
Figure 19 : Configuration Eclipse - Perspective ABAP	20
Figure 20 : Nouveau projet ABAP	20
Figure 21 : Nouveau projet ABAP - System Connection	21
Figure 22 : Nouveau projet ABAP - Connection Settings	21
Figure 23 : Nouveau projet ABAP - Logon to System	22
Figure 24 : Nouveau projet ABAP - Project Name and Packages	22
Figure 25 : Création d'une entité CDS	23
Figure 26 : Création entité CDS – ABAP Repository Object.....	23
Figure 27 : Création entité CDS - Data Definition	24
Figure 28 : Création entité CDS - Transport Request	24
Figure 29 : Création entité CDS - Template	25
Figure 30 : Vue CDS.....	25
Figure 31 : Résultat de la requête CDS	26
Figure 32 : Dictionary ABAP -Vue CDS	27
Figure 33 : Construction d'une vue CDS - Résumé	27
Figure 34 : Programme ABAP avec vue CDS	28
Figure 35 : Annotations CDS	28
Figure 36 : Jointure CDS	30
Figure 37 : Association CDS	30
Figure 38 : Association CDS - Réutilisation et filtre.....	31
Figure 39 : Association CDS - Programme ABAP	31
Figure 40 : Création d'un contrôle d'accès	31
Figure 41 : Contrôle d'accès - Templates	32
Figure 42 : Contrôle d'accès CDS	32
Figure 43 : Contrôle d'accès CDS - Autorisation	33
Figure 44 : Résultat avec le contrôle d'accès.....	33
Figure 45 : Résultat sans le contrôle d'accès.....	33
Figure 46 : Extension CDS	34
Figure 47 : Signe d'une extension existante	34
Figure 48 : Affichage du nom de l'extension	34
Figure 49 : Résultat de l'extension de vues	35
Figure 50 : Paramètres CDS	35
Figure 51 : Paramètres CDS - Insertion des valeurs	36

Figure 52 : Paramètres CDS - Résultat.....	36
Figure 53 : Paramètres CDS - Utilisation dans un programme ABAP	37
Figure 54 : Gestion des langues CDS	37
Figure 55 : Prototype HANA CDS - Droits de l'utilisateur.....	38
Figure 56 : Installation ADT Tools "Available Software" HANA	39
Figure 57 : Configuration Eclipse - Perspective HANA	40
Figure 58 : Configuration Eclipse - Nouveau système HANA	40
Figure 59 : Configuration Eclipse - Spécification du système HANA.....	40
Figure 60 : Configuration Eclipse - Propriétés de connexion HANA.....	41
Figure 61 : Configuration Eclipse - Repositories Workspace HANA.....	41
Figure 62 : Création d'un document CDS	42
Figure 63 : Création d'un document CDS - Sélection fichier.....	42
Figure 64 : Document CDS	43
Figure 65 : Documents CDS - Association	44
Figure 66 : Document CDS - Activation.....	44
Figure 67 : BD HANA - Tables	45
Figure 68 : Temps d'exécution Open SQL VS CDS View	46
Figure 69 : Annexe I - Tables de base de données	52
Figure 70 : Annexe I - Contenu des tables de la base de données	53
Figure 71 : Annexe II - Classe objet	54
Figure 72 : Annexe II - Rôle simple et autorisations.....	54
Figure 73 : Annexe III - Code completion	55
Figure 74 : Annexe III - Wizard pour les clauses « ON »	55
Figure 75 : Annexe III - Définition de la clause « ON »	56
Figure 76 : Annexe III - Editeur Graphique.....	56
Figure 77 : Annexe III - Active Annotations Views	57
Figure 78 : Annexe III - Active Annotations Views Details	57
Figure 79 : Annexe IV - Ajout d'un nouveau Script SQL.....	58
Figure 80 : Annexe IV - Script SQL	58
Figure 81 : Annexe IV - Open Data Preview	58
Figure 82 : Annexe IV – Données de la table	59
Figure 83 : Annexe IV - Format du fichier csv	59
Figure 84 : Annexe IV - Import Data from Local File	59
Figure 85 : Annexe IV - Import Data from Local File - Target System.....	60
Figure 86 : Annexe IV - Import Data from Local File - Define Import Properties.....	60
Figure 87 : Annexe IV - Import Data from Local File - Mappings	61
Figure 88 : Annexe IV - Import Data from Local File - Summary	62
Figure 89 : Annexe IV - Résultat	62
Figure 90 : Annexe VII - Détails des heures.....	65
Figure 91 : Annexe VIII - Rapport hebdomadaire.....	66

Liste des abréviations

ABAP	Advanced Business Application Programming
ACID	Atomicité, Cohérence, Isolation, Durabilité
ADBC	ABAP Database Connectivity
ADT	ABAP Development Tools for Eclipse
AMDP	ABAP Managed Database Procedure
CDS	Core Data Services
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
DCL	Data Control Language
DDIC	Data Dictionary
DDL	Data Definition Language
DSL	Domain Specific Language
EHP	Enhancement Packages
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
HANA	High Performance Analytics Appliance
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
OData	Open Data Protocol
OLAP	OnLine Analytical Processing
OLTP	OnLine Transaction Processing
OS	Operating System
PLM	Product Lifecycle Management
QL	Query Language
SAP	Systems Applications Products
SAPUI5	SAP User Interface for HTML 5
SCM	Supply Chain Management
SGBDR	Système de gestion de base de données relationnelle
SP	Support Package Stack
SPS	Service Pack Stack
SQL	Structured Query Language
SRM	Supplier Relationship Management
UI	User interface
URI	Uniform Resource Identifier

1. Introduction

A la fin du cursus de Bachelor de la filière Informatique de gestion auprès de la HES-SO Valais-Wallis, les étudiants développent un projet de fin d'études. Dans mon cas, ce travail a été réalisé sous la surveillance du Prof. Dr. Werner Maier et l'assistance de Monsieur Frédéric Morand.

Dans un premier temps, SAP CDS a été étudié sous un angle théorique afin de mettre en avant l'ensemble de ses fonctionnalités. L'étude rapporte notamment sur le concept de cette infrastructure unique, sur les différences entre ABAP CDS et HANA CDS, ainsi que sur les avantages de SAP CDS par rapport à l'*ABAP Dictionary*. Dans un deuxième temps, une fois l'approche théorique achevée, deux prototypes ont été développés afin d'illustrer de manière pratique les concepts étudiés.

Ce travail a débuté le 2 mai 2015 et s'est terminé le 31 juillet 2017. Les deux premiers mois d'écriture de ce travail se sont déroulés en parallèle avec les cours du cursus académique, puis, jusqu'à la fin juillet 2017, l'écriture du travail a été réalisée de manière exclusive. Ce travail était mon premier choix et m'a permis d'élargir mes connaissances dans le domaine SAP, domaine que je privilégie particulièrement pour mon avenir professionnel. Il m'a également permis de mettre en œuvre la méthodologie Agile apprise durant le cursus académique. Cette méthodologie a été adaptée dans le cadre du projet.

2. Technologies

Les technologies importantes abordées tout au long de ce rapport sont détaillées dans cette partie.

2.1. SAP HANA Platform

La plate-forme informatique SAP HANA (High Performance Analytics Appliance), dite *In-Memory*, prend en charge différentes applications pour les regrouper dans un système et permet de réimaginer les processus métier afin de fournir des informations en temps réel sur la base de données toujours à jour utilisée pour la Business Intelligence. En effectuant l'analyse des données en temps réel, elle permet de réduire les redondances, le matériel et les opérations informatiques (Karamba, 2017). Elle est l'une des premières plates-formes de gestion de données à gérer le traitement des transactions (OLTP pour OnLine transaction processing) et les analyses en mémoire (OLAP pour OnLine Analytical Processing) sur une seule copie de données. SAP HANA offre également la possibilité de développer des applications, d'intégrer des données et d'assurer leur bonne qualité. Elle permet ainsi de gérer un gros volume de données structurées ou non structurées. La plate-forme offre également la possibilité de déployer les données sur un site web ou le cloud (SAP, 2017).

Comme illustré par l'image ci-dessous, la plate-forme SAP HANA regroupe toutes les données venant de différentes applications dans une seule copie de données (Technology, 2017).

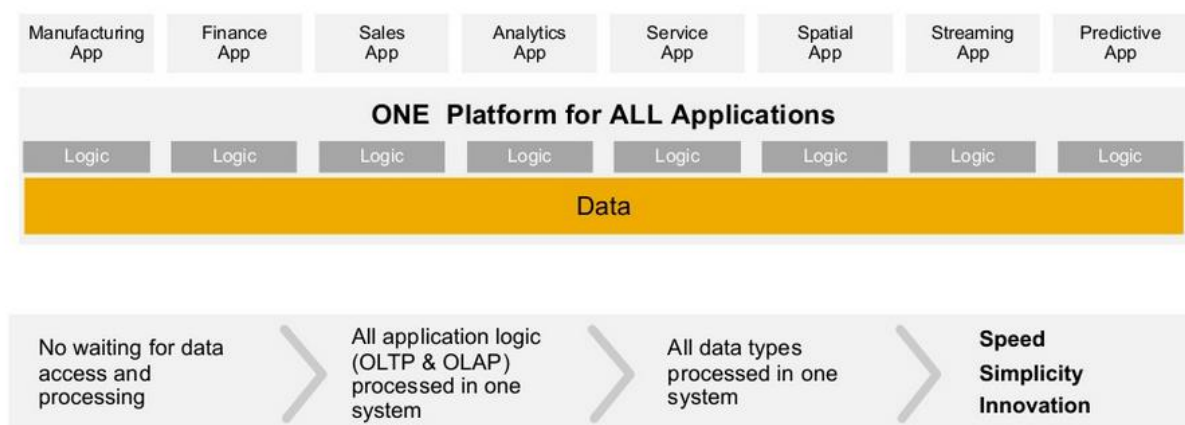


Figure 1: Plate-forme SAP HANA¹

¹ Source : <https://www.sap.com/documents/2016/02/bedb9729-5f7c-0010-82c7-eda71af511fa.html>

2.2. SAP HANA

SAP HANA correspond à la base de données relationnelle (SGBDR) en mémoire de SAP HANA Platform qui regroupe le traitement OLAP et OLTP sur une même base de données (SAP, 2017). Elle permet d'exécuter les analyses en même temps que les transactions sur des données à jour et ce, rapidement (Bizien, 2017). Elle possède les propriétés ACID, soit (Christophe, 2017) :

- **Atomicité** : une opération se fait soit en entier, soit n'est pas exécutée ;
- **Cohérence** : les données doivent être cohérentes ;
- **Isolation** : les opérations sont indépendantes les unes des autres ;
- **Durabilité** : les modifications sont enregistrées même en cas de problèmes tels qu'une coupure d'électricité.

Construite en colonnes dans la mémoire, elle optimise le traitement multidimensionnel. Avec les différents outils de modélisation et la bibliothèque de fonctions, elle offre une exécution directement sur la base de données, évitant ainsi des latences dues au déplacement de données entre une base de données et un serveur d'application. Il faut noter que les différentes bases de données condensées d'une seule et unique base gardent tout de même un isolement l'une de l'autre. Les tables, construites sur plusieurs étages, permettent de conserver en mémoire les données déjà consultées (Bizien, 2017).

Données en ligne			
Identifiant	Nom	Prénom	Salaire
1	Durant	Jacques	40000
2	Dupont	Marie	50000
3	Martin	Jeanne	44000
4	Sia	Marie	40000

Données en colonnes				
Nom	Durant : 1	Dupont : 2	Martin : 3	Sia : 4
Prénom	Jacques : 1	Marie : 2	Jeanne : 3	Marie : 4
Salaire	40000 : 1	50000 : 2	44000 : 3	40000 : 4

Données en colonnes optimisées				
Nom	Dupont : 2	Durant : 1	Martin : 3	Sia : 4
Prénom	Jacques : 1	Jeanne : 3	Marie : 2,4	
Salaire	50000 : 2	44000 : 3	40000 : 1,4	

Figure 2 : Données en ligne / Données en colonnes²

² Source : <http://pdvsap.blogspot.ch/2014/06/quest-ce-que-hana.html>

2.3. S/4HANA

S/4HANA est la nouvelle ERP de SAP, soit la quatrième version de SAP Business Suite, construite sur la plate-forme SAP HANA et fonctionnant uniquement avec la base de données HANA (SAP, 2017). Elle regroupe la suite SAP Business Suite en un seul produit dit en mémoire (Illumiti, 2017). Elle est conçue de manière plus simple, avec de nouvelles fonctionnalités. Elle est également plus rapide et offre une meilleure disponibilité pour le déploiement (site, cloud ou hybride). Son design est basé sur l'expérience utilisateur de SAP Fiori, laquelle est basée sur SAPUI5 (SAP User Interface for HTML5). S/4HANA permet par exemple de prédire ce que les clients seront susceptibles de vouloir afin d'améliorer les prévisions (Rousse, 2017).

Les principales simplifications et les nouvelles innovations offertes par S/4HANA sont :

- Les modèles de données ;
- L'expérience utilisateur ;
- La prise de décision ;
- Les processus métier ;
- L'Internet of Things ;
- Le Big Data ;
- Les réseaux d'entreprise.

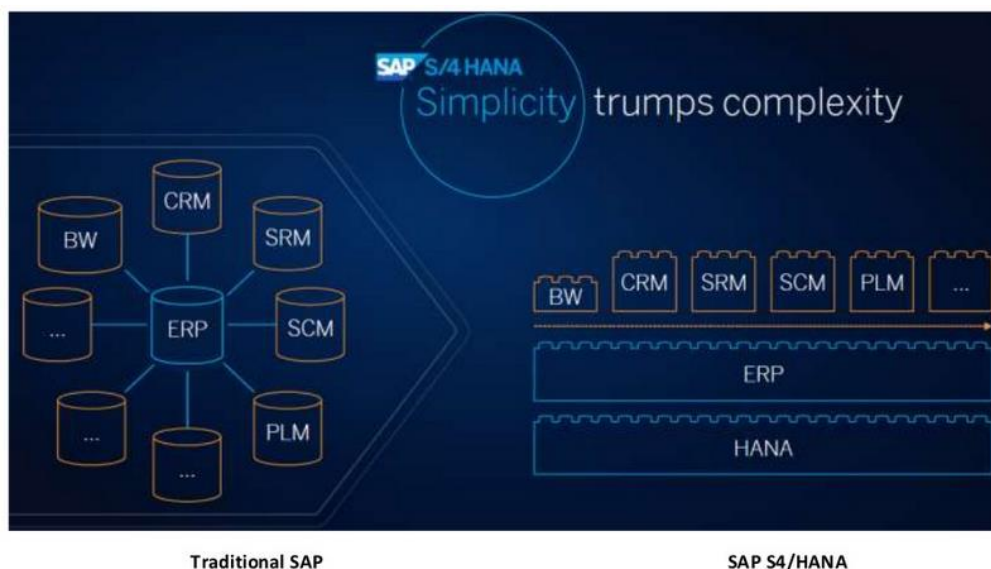


Figure 3: SAP vs SAP S/4HANA³

³ Source : <https://fr.slideshare.net/DebajitBanerjee/an-overview-of-sap-s4hana-64499419>

2.4. ABAP Dictionary

ABAP (Advanced Business Application Programming), se traduisant par programmation avancée des applications métier, est un langage de programmation développé par SAP. *ABAP Dictionary* contient la définition de tous les objets SAP se trouvant dans l'entrepôt de données (repository). Ce dernier contient l'intégralité du code source et des objets de développement enregistrés dans la base de données. Ces données sont indépendantes du mandant et subdivisées en fonction des composantes applicatives (ABAP Dictionary, communication personnelle, support de cours SAP).

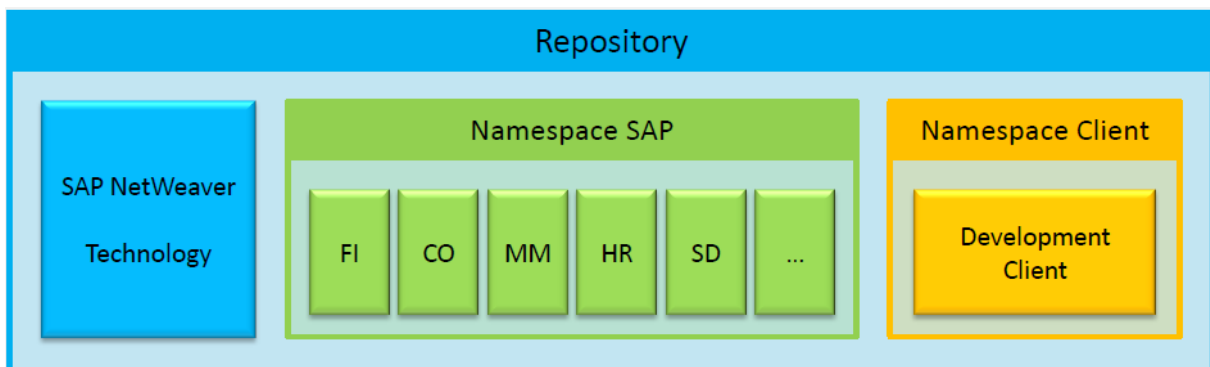


Figure 4 : Repository SAP⁴

Le dictionnaire ABAP permet de centraliser tous les éléments se trouvant dans la base de données SAP et favorise une utilisation dans le système sans redondances. Tous les éléments créés, modifiés ou supprimés sont automatiquement mis à jour dans la base de données. Ceux-ci peuvent être des tables, des vues, des traductions ou de la documentation. Ainsi, l'intégrité des données, leur cohérence et la sécurité sont assurées (ABAP Dictionary, communication personnelle, support de cours SAP).

Le dictionnaire ABAP est intégré dans l'ABAP Workbench. Le système SAP fonctionnant de manière interprétative, il permet au dictionnaire ABAP d'être intégré dans l'environnement de développement. Les interprètes ne perçoivent pas les objets originaux mais des représentations internes de ces derniers. Lorsqu'une modification est effectuée dans le dictionnaire ABAP, le système met à jour ces représentations internes (SAP, ABAP Dictionary, 2017).

⁴ Source : Support du cours SAP « 0106_ABAP_Transport_v5.pdf »

2.5. Open SQL

L'Open SQL est un ensemble d'instructions permettant d'interroger la base de données. Il est l'équivalent du SQL, simplifié et basé sur une syntaxe et une sémantique uniformes pour tous les systèmes de base de données pris en charge par SAP. Ainsi, les programmes ABAP utilisant ces instructions seront compatibles avec n'importe quel système SAP, indépendamment du système de base de données utilisé (Guru99, Open SQL & Native SQL in SAP ABAP, 2017). Les instructions permettent d'accéder aux tables de base de données déclarées dans le dictionnaire ABAP.

Comme le démontre l'image ci-dessous, avec son évolution, l'Open SQL se rapproche du SQL standard.

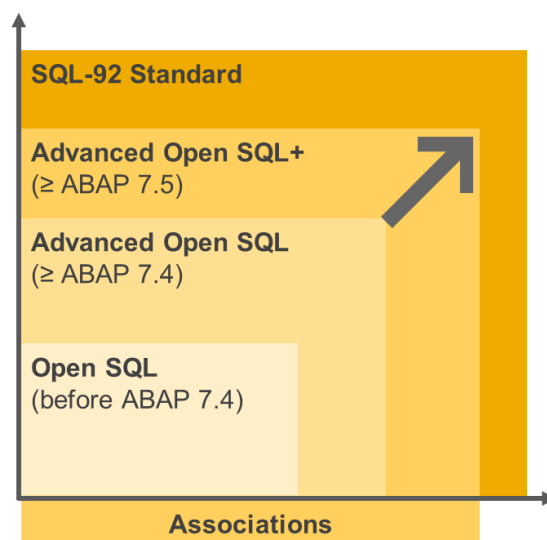


Figure 5 : Evolution Open SQL⁵

2.6. OData

L'OData, pour Open Data Protocol, comme l'indique son nom, est un protocole. Celui-ci permet de partager, de créer et de consommer des données venant de différentes sources à travers le web. Ce protocole étant basé sur les technologies HTTP, Atom et JSON, différents services et applications peuvent consommer les données mises à disposition à travers une URI (Uniform Resource Identifier) (Casabianca, 2017).

⁵ Source : https://blogs.sap.com/wp-content/uploads/2015/10/a4h_05_enhancedopensql_814086.png

3. SAP CDS

SAP CDS correspond à la nouvelle génération pour la définition des données, conçu pour les applications centrées sur la base de données. Le *Core Data Services* est une couche d'abstraction multiplateforme, qui peut être comparée à l'OData pour l'interface utilisateur (TECHED, 2017).

Intégré à la gestion du cycle de vie de la plate-forme SAP et basé sur le SQL natif, SAP CDS offre une plus grande productivité à travers un maximum de transparence pour fonctionner avec différents modèles de programmation (SAP, SAP netweaver application server for ABAP 7.5, 2017).

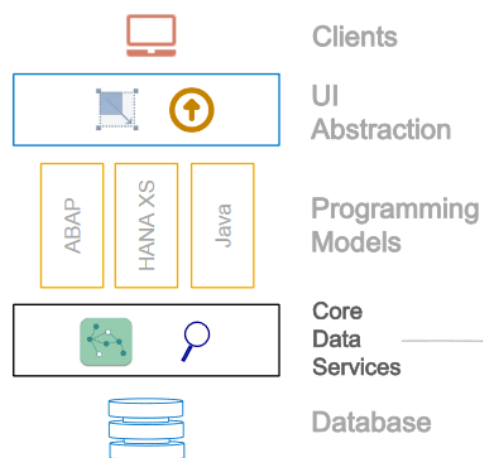


Figure 6 : Core Data Services⁶

3.1. Concept

Techniquement, il s'agit d'une amélioration de SQL permettant de fournir un langage dédié (DSL, Domain Specific Language), et ce, afin de définir des tables et des vues de base de données sémantiquement riches, correspondant aux entités CDS (Bisht, 2017).

Le CDS est donc composé des langages suivants (TECHED, 2017) :

- **Data Definition Language (DDL)**, pour la modélisation et l'extraction des données. Il est basé sur le SQL natif mais avec une productivité plus élevée ;
- **Query Language (QL)**, pour la lecture des données en utilisant les entités CDS à travers Open SQL dans l'ABAP ;
- **Data Control Language (DCL)**, pour la gestion des autorisations.

⁶ Source : <https://assets.cdn.sap.com/sapcom/docs/2016/02/bedb9729-5f7c-0010-82c7-eda71af511fa.pdf>

Comme pour le DDIC (*Data Dictionary = ABAP Dictionary*), les modèles de données fondés sur le CDS sont utilisés pour interagir avec la base de données. Les artefacts CDS sont stockés dans le DDIC et sont accessibles depuis un programme ABAP de la même manière que pour les tables et les vues ordinaires (Ahmed, 2017).

Conçu à la base pour le développement d'applications sur SAP HANA, il a ensuite été intégré à l'ABAP (se référer au point suivant pour connaître les prérequis minimaux pour l'utilisation du CDS) (Kessler, 2017).

Comme démontré par l'image ci-dessous, le CDS propose une amélioration ou de nouvelles fonctions et expressions mises en avant à travers un texte en gras (SAP, SAP netweaver application server for ABAP 7.5, 2017).

Generic SQL Expressions	Conversion Functions	String Functions	Arithmetic Functions	Date and Time Functions
Simple- / Searched-Case Coalesce	Unit Conversion Currency Conversion Cast Hex2Bin Bin2Hex Decimal Shift	Concat Concat with Space Instr Left / Right Length Lpad / Rpad Ltrim / Rtrim Replace	Abs Ceil Div (Integer) Division (Dec) Floor Mod Round	Days between Add Days Add Month Current Tmstp Date is Valid Tmstp is Valid Seconds btw. Tmstps Add seconds

Figure 7 : Expressions et fonctions CDS⁷

En plus de ces nouveautés ou améliorations, les deux nouvelles fonctionnalités majeures proposées sont les annotations et la gestion des autorisations, qui seront expliquées au chapitre quatre.

3.2. Prérequis

Même si SAP CDS était initialement conçu pour SAP HANA, il reste toutefois disponible à partir des plates-formes SAP suivantes (TECHED, 2017) :

- SAP NetWeaver 7.4 SP05 (Support Package Stack 05);
- SAP HANA SPS6 (Service Pack Stack);
- SAP Business Suite EHP7, suite on HANA (Enhancement Packages) ;
- S/4 HANA ;
- SAP Business Warehouse 7.3.

⁷ Source : <https://www.sap.com/documents/2016/02/782ac129-5f7c-0010-82c7-eda71af511fa.html>

3.3. ABAP CDS

L'ABAP CDS permet, à travers un *framework*, la définition et l'utilisation de modèles de données directement sur la base de données. Ces modèles, basés sur le langage de définition des données et le langage de contrôle de données, se nomment des entités (SAP, SAP - ABAP CDS Development User Guide, 2017).

Il existe deux types d'entités ABAP CDS :

1. **Les vues** : comme les données d'une application se trouvent sur différentes tables de la base de données, les vues CDS permettent de réorganiser les champs de tables selon les différents besoins. Une vue CDS peut être définie pour des tables ou des vues déjà existantes dans la base de données, mais également pour d'autres vues CDS déjà créées. Lorsqu'une vue est activée, une entité CDS et une vue SQL sont créées dans le dictionnaire ABAP (SAP, SAP - ABAP CDS Development User Guide, 2017) ;
2. **Les fonctions de tables** : elles correspondent aux fonctions de tables existantes sur la base de données et peuvent être utilisées dans le CDS (SAP, SAP - ABAP CDS Development User Guide, 2017).

3.4. HANA CDS

SAP a introduit le CDS avec SAP HANA dans le but de définir et de consommer les données directement sur la base de données plutôt que sur le serveur. A la base, le CDS n'était disponible que sur SAP HANA SPS06, puis a été intégré à partir de SAP NetWeaver 7.4 SP05 (Kessler, 2017). C'est pourquoi les spécifications sont pratiquement identiques à celles de l'ABAP CDS à la différence que les sources DDL de HANA ne sont pas utilisables telles quelles avec ABAP CDS ni réciproquement (Keller, CDS – One Concept, Two Flavors, 2017).

Avec SAP HANA, le CDS offre la possibilité « de définir les artefacts qui composent le modèle de persistance des données » (Portal, 2017). A travers la création d'un document CDS, le DDL permet de définir les artefacts et les éléments suivants (Portal, 2017) :

- Des entités (tables) ;
- Des vues ;
- Des types de données définis par l'utilisateur ;
- Des contextes ;
- Des associations ;
- Des annotations.

Les documents CDS possèdent le format *.hdbdd* et peuvent être considérés comme une alternative aux *.hdbtable* et *.hdbstructure* (Jung, 2017).

3.5. Différences entre ABAP CDS & HANA CDS

Le CDS ayant été introduit à la base avec SAP HANA, les spécifications sont pratiquement identiques mais les DDL de HANA ne sont pas réutilisables telles quelles avec ABAP CDS ni réciproquement (Keller, CDS – One Concept, Two Flavors, 2017).

Le CDS basé sur SAP HANA fonctionne uniquement sur SAP HANA tandis que le CDS basé sur ABAP fonctionne sur la plupart des plate-formes, ainsi que sur SAP HANA (Kessler, 2017).

ABAP CDS et HANA CDS possèdent le même objectif, à savoir « représenter les définitions de données centrales comme une base commune pour le développement d'applications » (Kessler, 2017).

Quand devrait-on utiliser ABAP CDS et quand devrait-on privilégier HANA CDS ? Dans son article « ABAP on SAP HANA. Part IV. Core Data Services », Mohsin Ahmed répond à cette question de la façon suivante (Ahmed, 2017) :

1. Si l'exécution de SAP HANA se fait sans la pile ABAP, l'utilisation de l'ABAP CDS n'est pas possible et il faut utiliser HANA CDS ;
2. Si l'utilisation de la base de données HANA se fait avec une pile ABAP, les scénarii suivants sont possibles :
 - a. Si l'on souhaite utiliser les entités CDS en tant que types de données ou tester les annotations CDS dans ABAP, il faut utiliser ABAP CDS ;
 - b. Si l'on ne souhaite pas utiliser les entités CDS dans ABAP, mais que l'on veut les transporter comme des objets de l'ABAP repository, il est possible d'utiliser ABAP CDS ;
 - c. Si l'on ne souhaite pas utiliser les entités CDS en tant que types de données dans ABAP ou dans Open SQL, il est possible d'utiliser HANA CDS qui est mieux intégré dans SAP HANA. Un accès depuis ABAP est alors possible en utilisant SQL grâce à *ABAP Database Connectivity* (ADBC) et *ABAP Managed Database Procedure* (AMDP).

4. Fonctionnalités SAP CDS

Les différentes fonctionnalités offertes par SAP CDS sont détaillées dans cette partie exclusivement théorique. La mise en pratique est effectuée dans la partie prototype ABAP CDS.

4.1. Jointures

Les jointures permettent de regrouper des champs venant de deux ou plusieurs tables différentes. Les jointures disponibles avec le CDS sont les suivantes : *INNER JOIN* ou *JOIN*, *LEFT OUTER JOIN*, *RIGHT OUTER JOIN*. Les commandes *UNION* et *UNION ALL* sont également accessibles. Cette fonctionnalité est disponible à partir de SAP NetWeaver 7.4 SP05 (Swanepoel, New Data Modeling Features in SAP NW ABAP 7.4 SP5, 2017).

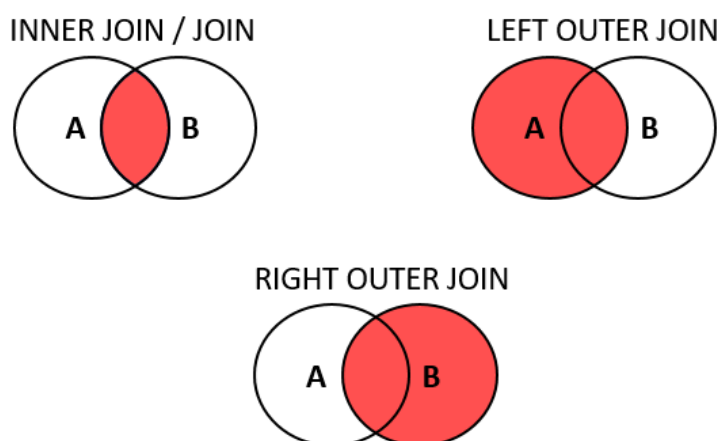


Figure 8 : Jointures⁸

4.2. Associations

Les associations CDS peuvent être considérées comme des jointures, du fait que lors de l'activation de la vue CDS, les associations sont converties en expressions *JOIN*. Elles ont la particularité d'être réutilisables et permettent de définir des relations entre la vue CDS actuelle et une autre source de données qui peut être par exemple une vue classique ou une entité CDS. Les relations sont définies à l'aide des cardinalités en utilisant la syntaxe *ASSOCIATION [MIN..MAX]* qui est par défaut fixée à *[0..1]* si aucune cardinalité n'est définie. Une autre particularité des associations est la possibilité d'utiliser des expressions de filtre avec une cardinalité (TECHED, 2017).

⁸ Source : Donnée de l'auteur

L'intérêt d'utiliser les associations à la place des jointures provient, d'une part, de la possibilité de définir les cardinalités et, d'autre part, de la simplification de lecture car « elle se rapproche de la pensée conceptuelle » (Santhosh, 2017). Il est important de retenir que, par défaut, les associations utilisent la jointure *LEFT OUTER JOIN*. Comme pour la jointure, elles sont disponibles à partir de SAP NetWeaver 7.4 SP05 (Swanepoel, New Data Modeling Features in SAP NW ABAP 7.4 SP5, 2017).

4.3. Annotations

Les annotations sont définies à l'aide du caractère « @ » dans le document CDS. Elles définissent les propriétés et le comportement au moment de l'exécution, comme par exemple la manière de sauvegarder les vues, le nom de la vue, la dépendance du client ou la référence entre les colonnes. Les annotations permettent d'ajouter des métadonnées (données décrivant des autres données) à l'entité CDS (Swanepoel, New Data Modeling Features in SAP NW ABAP 7.4 SP5, 2017).

Les annotations SAP CDS peuvent être soit des annotations ABAP, soit des annotations spécifiques à des *frameworks* (SAP, ABAP CDS - Annotations , 2017). Le détail pour chacune d'elles peut être consulté sous le lien suivant :

<https://help.sap.com/viewer/cc0c305d2fab47bd808adcad3ca7ee9d/7.51.1/en-US/630ce9b386b84e80bfade96779fbaeec.html> .

4.3.1. Annotations ABAP

Ces annotations sont évaluées par l'environnement d'exécution ABAP et sont disponibles à partir de SAP NetWeaver 7.4 SP05 (Swanepoel, New Data Modeling Features in SAP NW ABAP 7.4 SP5, 2017) :

- AbapCatalog ;
- AccessControl ;
- ClientDependent ;
- ClientHandling ;
- DataAging ;
- EndUserText ;
- Environment ;
- MappingRole ;
- Metadata ;
- Semantics.

4.3.2. Annotations spécifiques à des *frameworks*

Ces annotations sont évaluées durant l'exécution par des *frameworks* spécifiques tels que le détaille l'image suivante. Ces annotations sont disponibles dès la version SAP NetWeaver 7.5 (Keller, ABAP News for Release 7.50 – Annotations in ABAP CDS, 2017) :

- Analytics ;
- AnalyticsDetails ;
- Consumption ;
- DefaultAggregation ;
- EnterpriseSearch ;
- Hierarchy ;
- ObjectModel ;
- OData ;
- Search ;
- Semantics ;
- UI ;
- VDM.

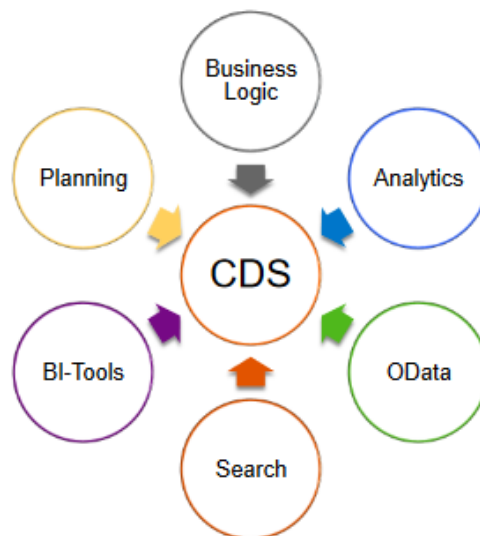


Figure 9 : Annotations CDS⁹

⁹ Source : <https://www.sap.com/documents/2016/02/782ac129-5f7c-0010-82c7-eda71af511fa.html>

4.4. Autorisations

Le CDS possède son propre concept d'autorisations à travers le DCL. Le contrôle des autorisations se fait à travers les conditions définies dans le CDS et celles définies dans les autorisations classiques (PFCG). Pour chaque vue CDS, il est possible de définir un contrôle d'accès à l'aide d'un DCL en lien avec les autorisations existantes dans SAP. Si aucune autorisation DCL n'est créée pour une entité CDS, celle-ci sera accessible par tous les utilisateurs (SAP, SAP - ABAP CDS Development User Guide, 2017). Cette fonctionnalité a été introduite à partir de la version SAP NetWeaver 7.40 SP10 mais n'a réellement été fonctionnelle qu'à partir de la version 7.50 (Keller, CDS – One Concept, Two Flavors, 2017).

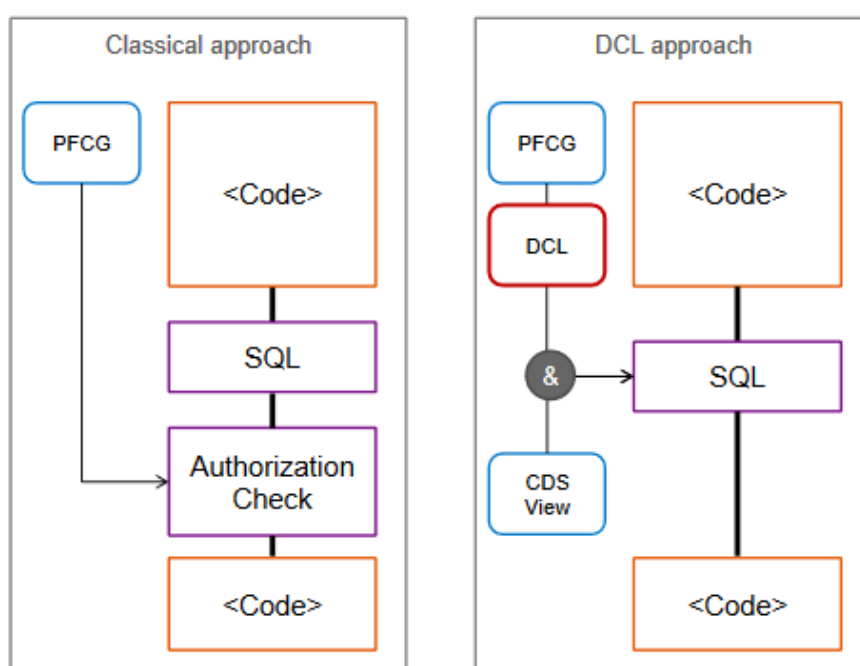


Figure 10 : Autorisations CDS¹⁰

¹⁰ Source : <https://www.sap.com/documents/2016/02/782ac129-5f7c-0010-82c7-eda71af511fa.html>

4.5. Extension de vues

L'extension de vues est basée sur le concept hiérarchique de « vue sur vue » afin d'optimiser un ensemble de résultats en transférant un minimum de données. Grâce à ces extensions de vues, il est par exemple possible d'ajouter des colonnes supplémentaires, des expressions ou des associations à une vue existante (TECHED, 2017). Cette fonctionnalité est disponible à partir de la version Netweaver 7.40 SP08 (Keller, ABAP News for 7.40, SP08 – ABAP Core Data Services (CDS), 2017).

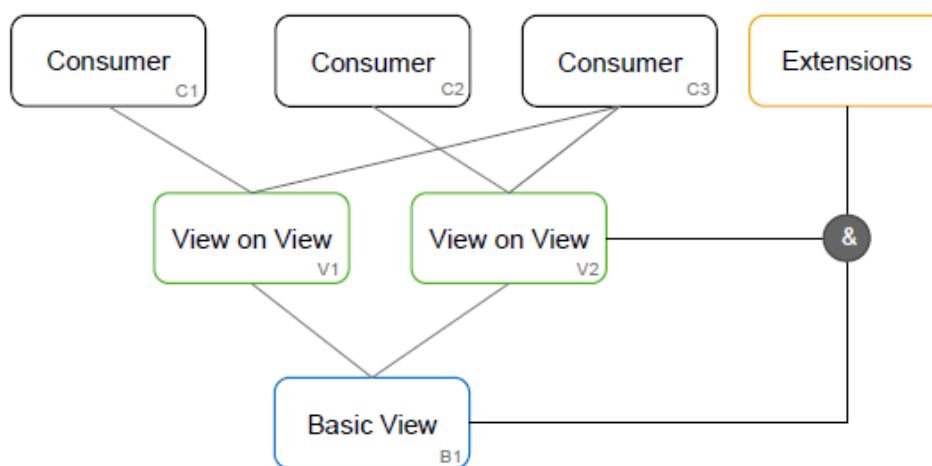


Figure 11 : Extension de vues CDS¹¹

¹¹ Source : <https://assets.cdn.sap.com/sapcom/docs/2016/02/bedb9729-5f7c-0010-82c7-eda71af511fa.pdf>

5. Prototype ABAP CDS

La réalisation du prototype a été réalisée sur la version d'Eclipse Neon.3 Release (4.6.3) et les versions SAP ERP 6.08 (SAP ERP 6.0 EHP8) et SAP NetWeaver 7.5 SP06. Il a été convenu avec Monsieur Frédéric Morand de créer de nouvelles tables pour les tests, lesquelles sont détaillées dans les annexes. Ces tables simulent des commandes clients d'instruments ou d'accessoires musicaux.

5.1. Configuration de l'environnement Eclipse

Cette partie permet de configurer l'environnement de développement Eclipse en y ajoutant les outils de développement nécessaires, la connexion au système SAP et la création d'un nouveau projet.

5.1.1. Outils de développement ABAP

Afin de pouvoir utiliser l'environnement de développement ABAP sur Eclipse, il est nécessaire d'ajouter le plug-in de développement ABAP. Les étapes pour l'installation de l'ADT (ABAP Development Tools) sont les suivantes (olgadolinskaja, 2017) :

1. Télécharger un environnement de développement Eclipse ;
2. Une fois l'environnement installé, ouvrir Eclipse et utiliser le menu : *Help > Install New Software* ;

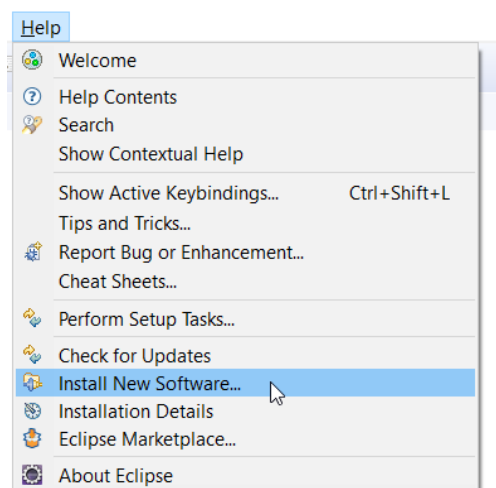


Figure 12 : Installation ADT Tools - Install New Software¹²

¹² Source : Donnée de l'auteur sur Eclipse Neon

3. Sélectionner le bouton *Add* pour insérer l'URL ;

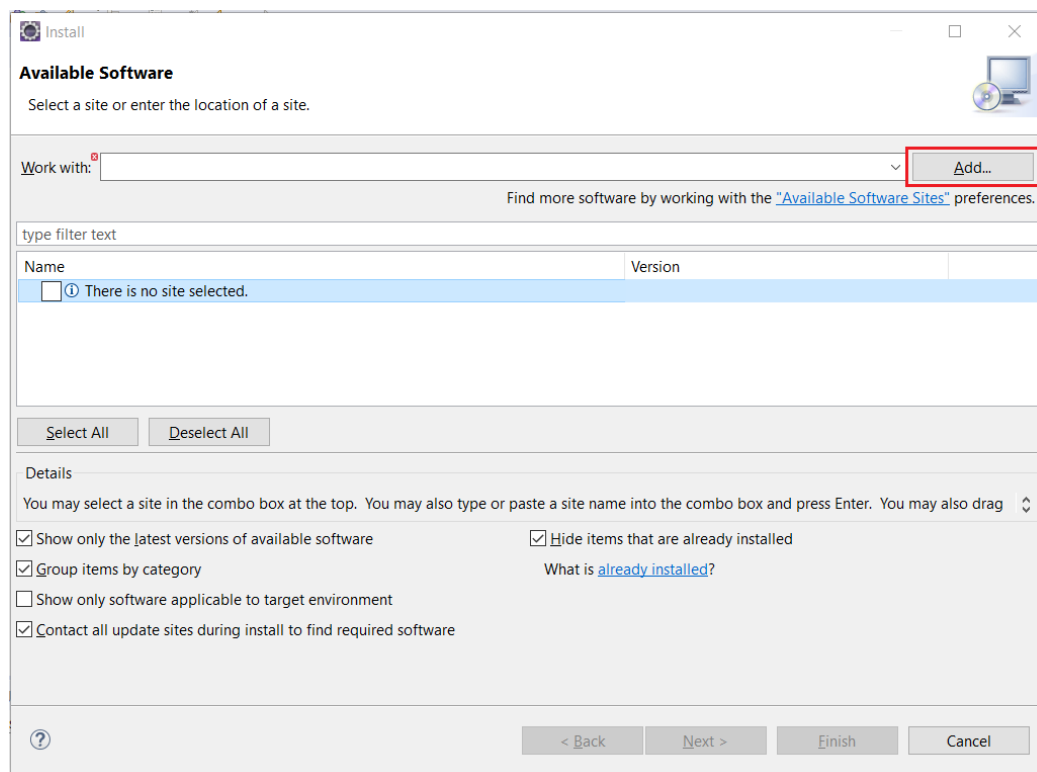


Figure 13 : Installation ADT Tools - Add link¹³

4. Dans la fenêtre *Add Repository*, insérer par exemple comme nom « Neon » et comme URL «<https://tools.hana.ondemand.com/neon> » ;

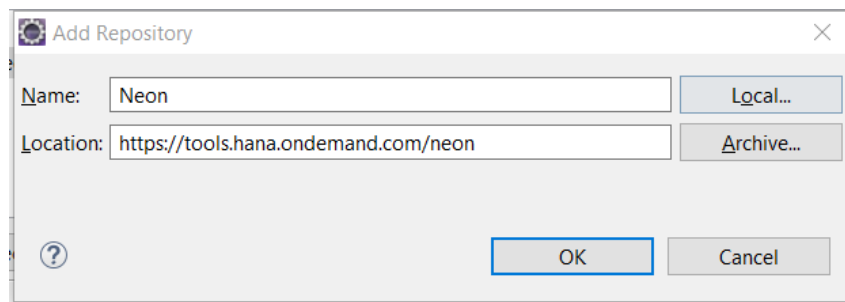


Figure 14 : Installation ADT Tools - Add Repository¹⁴

¹³ Source : Donnée de l'auteur sur Eclipse Neon

¹⁴ Source : Donnée de l'auteur sur Eclipse Neon

5. Dans la liste des outils de développement disponibles, sélectionner « *ABAP Development Tools for SAP NetWeaver* » ;

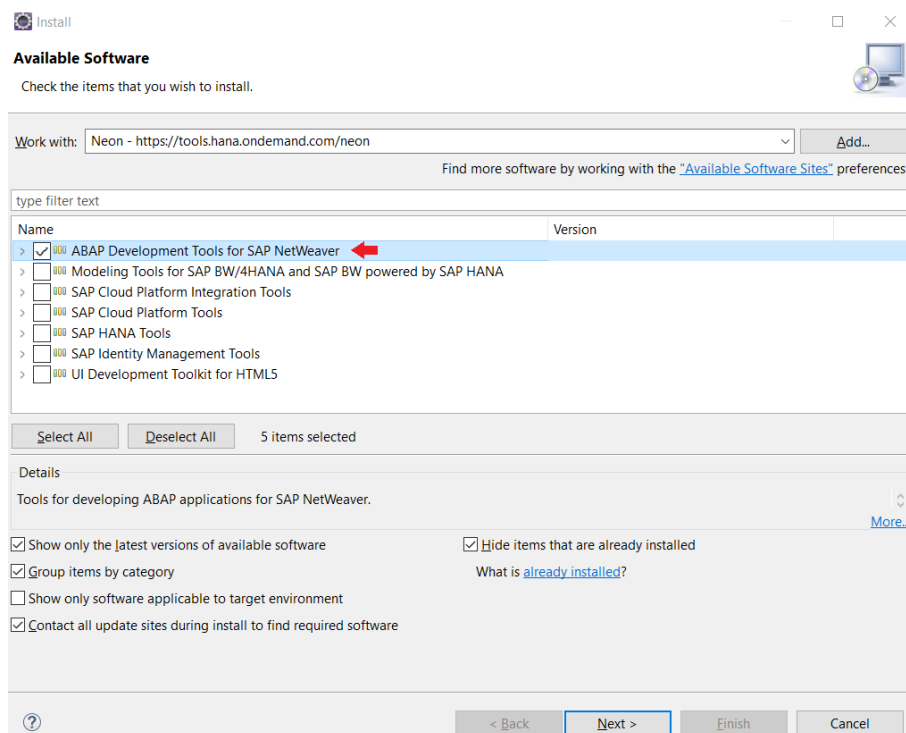


Figure 15 : Installation ADT Tools - Available Software¹⁵

6. Dans la fenêtre *Install Details*, vous pouvez visualiser les éléments qui seront installés ;

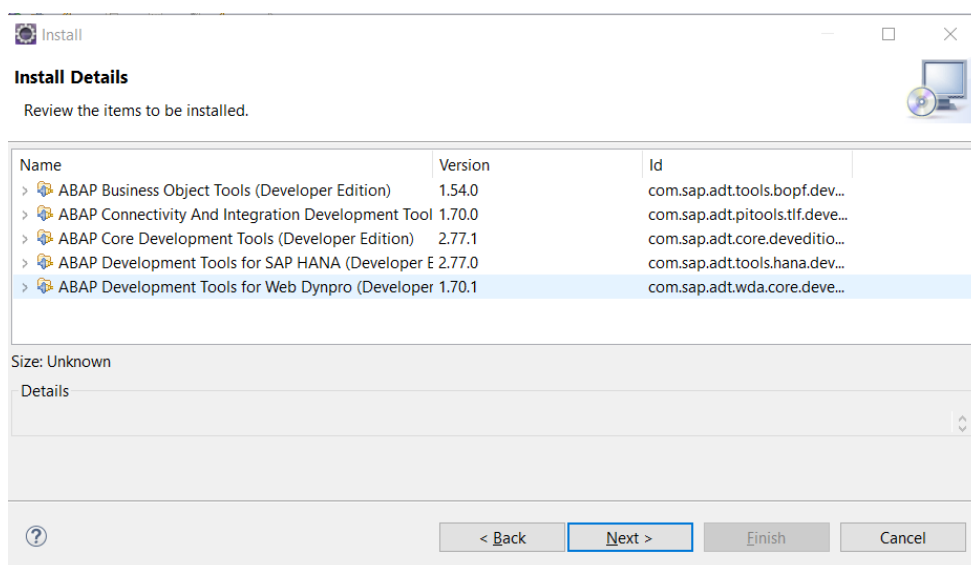


Figure 16 : Installation ADT Tools - Review Items¹⁶

¹⁵ Source : Donnée de l'auteur sur Eclipse Neon

¹⁶ Source : Donnée de l'auteur sur Eclipse Neon

7. Dans la fenêtre *Review Licenses*, la dernière étape consiste à accepter les termes du contrat de licence ;

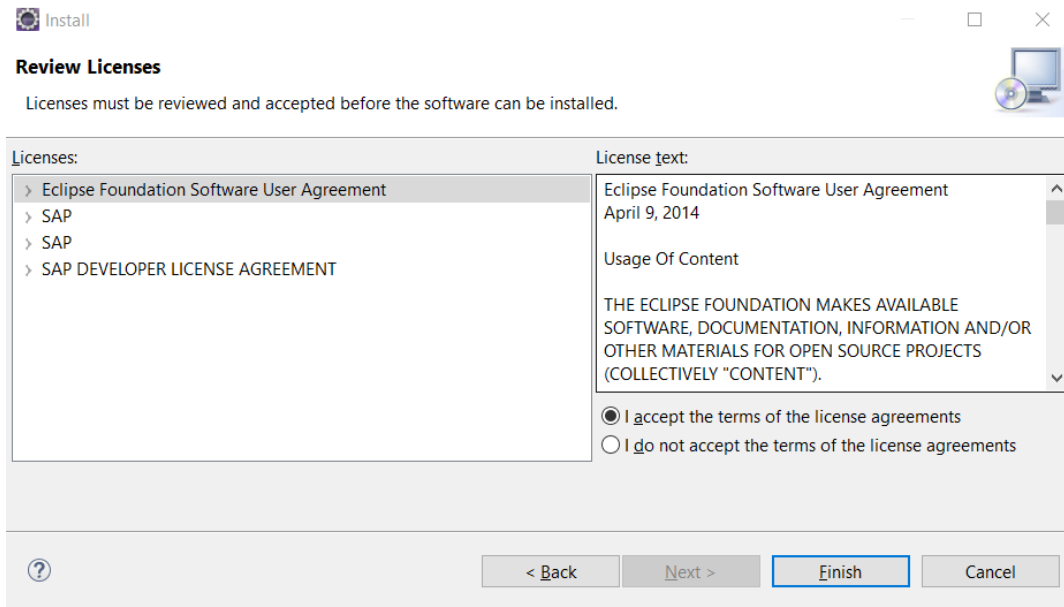


Figure 17 : Installation ADT Tools - Review Licenses¹⁷

8. Une fois l'installation terminée, redémarrer Eclipse.

5.1.2. Nouveau projet ABAP

Avant de créer un nouveau projet ABAP, il est nécessaire de modifier la perspective JAVA en ABAP afin d'avoir accès à tous les objets de développement :

1. Dans le menu, sélectionner : *Window > Perspective > Open Perspective > Other* ;

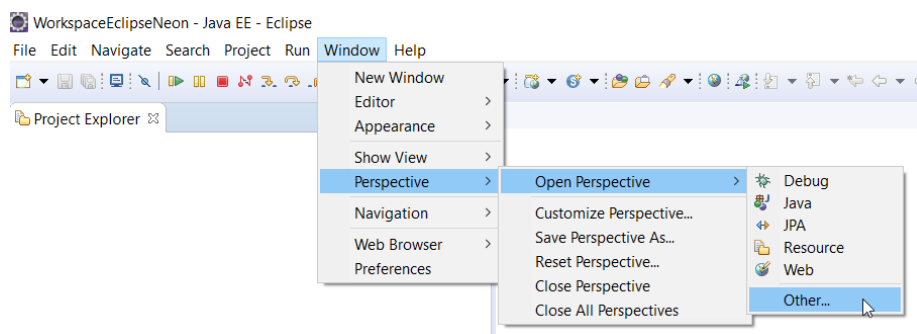


Figure 18 : Configuration Eclipse - Modification perspective¹⁸

¹⁷ Source : Donnée de l'auteur sur Eclipse Neon

¹⁸ Source : Donnée de l'auteur sur Eclipse Neon

2. Dans la liste des perspectives disponibles, sélectionner « ABAP ».

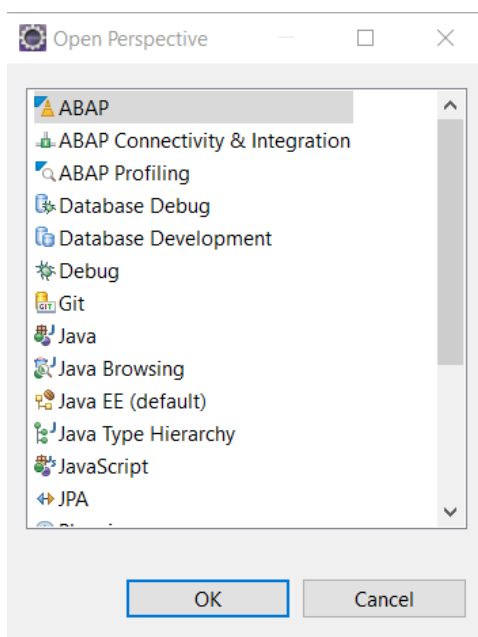


Figure 19 : Configuration Eclipse - Perspective ABAP¹⁹

Il est maintenant possible de créer un nouveau projet ABAP, en utilisant le chemin suivant : *File > New > ABAP Project*.

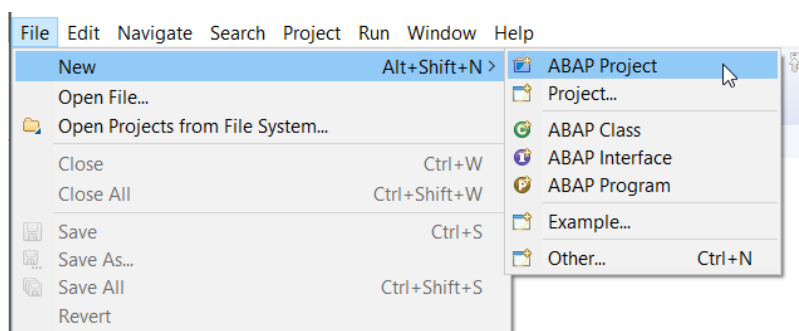


Figure 20 : Nouveau projet ABAP²⁰

Le premier écran permet de sélectionner une connexion SAP pour lier le projet ABAP.

¹⁹ Source : Donnée de l'auteur sur Eclipse Neon

²⁰ Source : Donnée de l'auteur sur Eclipse Neon

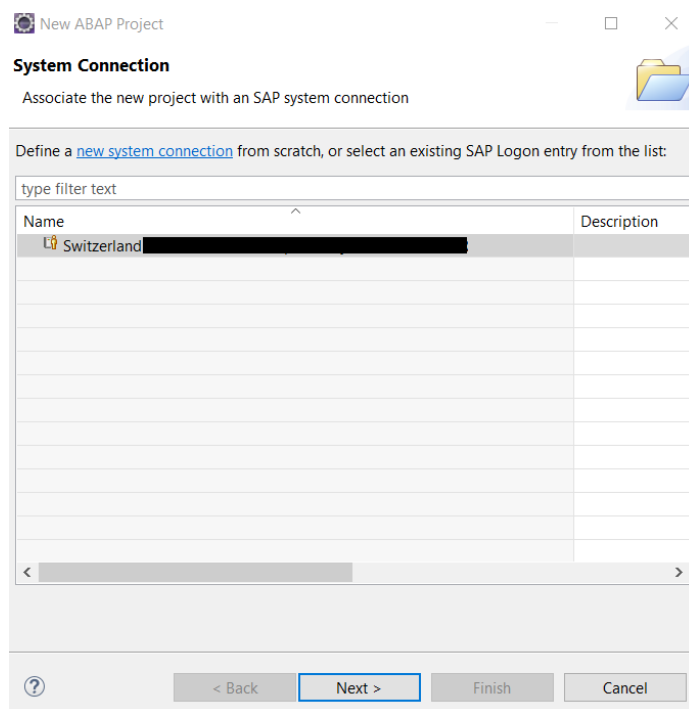


Figure 21 : Nouveau projet ABAP - System Connection²¹

Les paramètres de connexion sont automatiquement récupérés en fonction du système préalablement sélectionné.

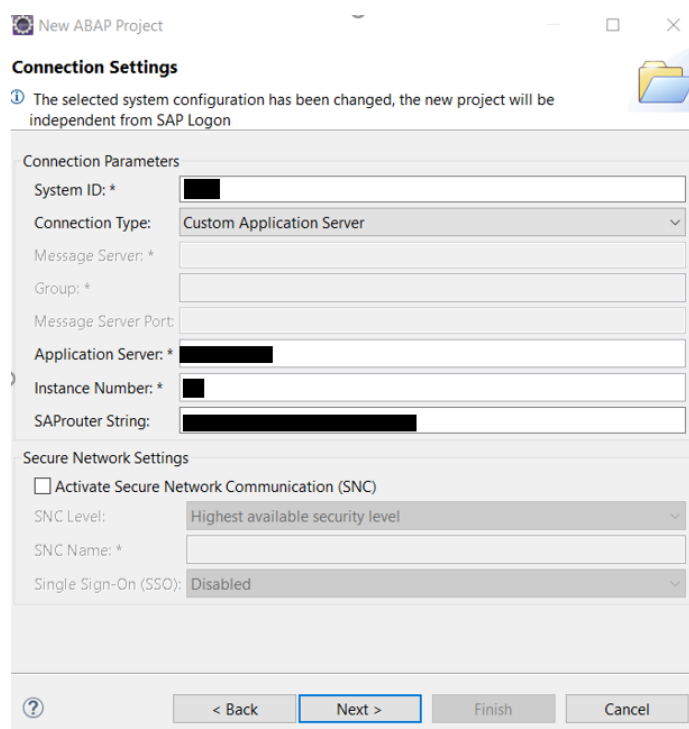


Figure 22 : Nouveau projet ABAP - Connection Settings²²

²¹ Source : Donnée de l'auteur sur Eclipse Neon

²² Source : Donnée de l'auteur sur Eclipse Neon

L'écran suivant requiert d'insérer les informations relatives à la connexion SAP.

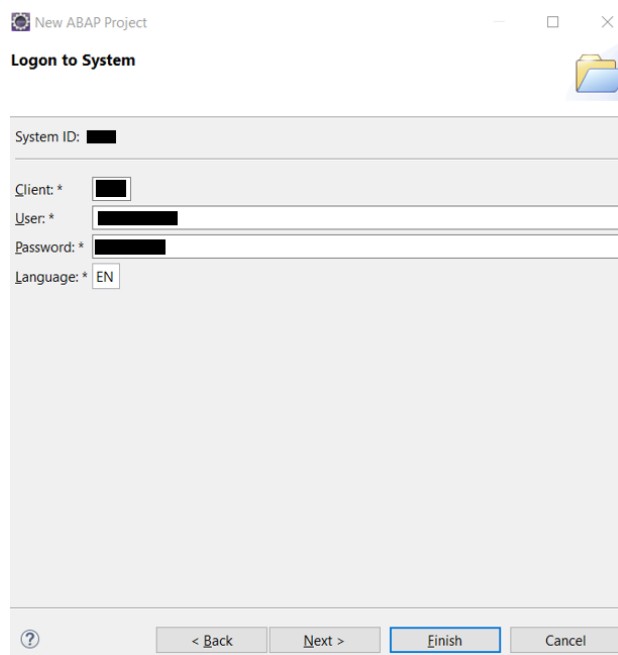


Figure 23 : Nouveau projet ABAP - Logon to System²³

Le nom du projet est alors entré et un package existant sur SAP doit lui être assigné à l'aide du bouton *Add*.

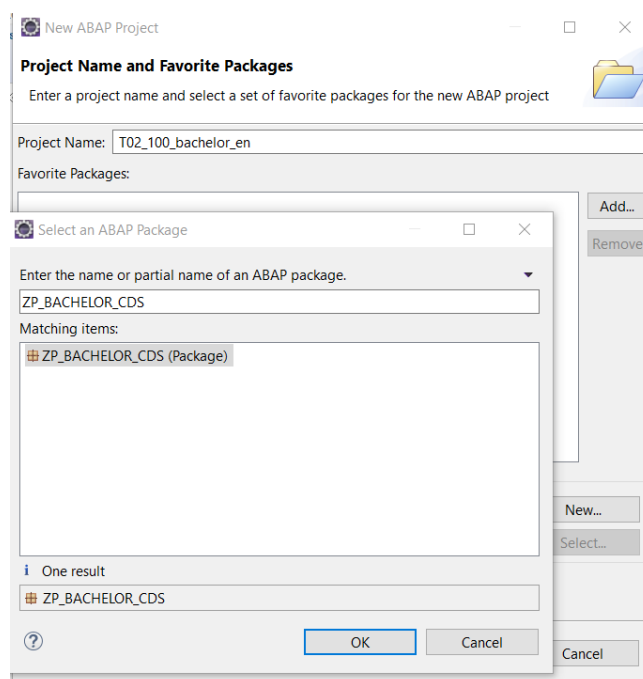


Figure 24 : Nouveau projet ABAP - Project Name and Packages²⁴

²³ Source : Donnée de l'auteur sur Eclipse Neon

²⁴ Source : Donnée de l'auteur sur Eclipse Neon

5.2. Création d'une entité CDS

La création des entités CDS s'exécute de la manière suivante (D028018, 2017) (Admin, 2017) :

1. Dans le projet ABAP, sélectionner le nœud du sommet, faire un clic droit et suivre le chemin : *New > ABAP Repository Object* ;

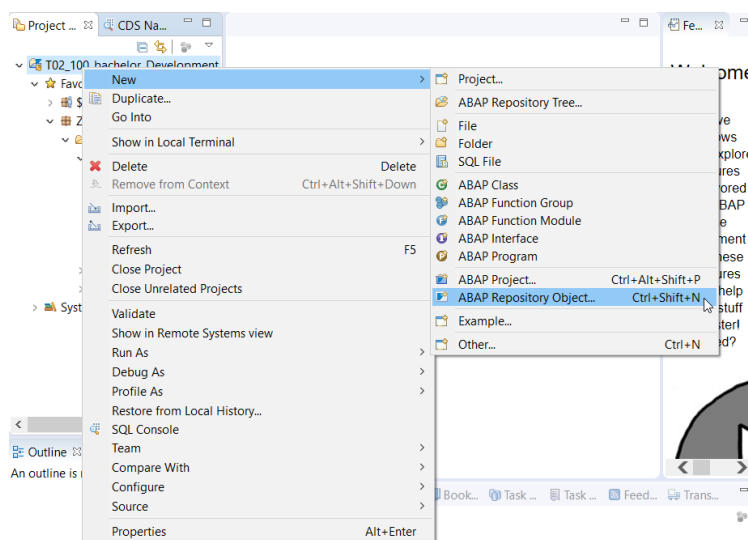


Figure 25 : Création d'une entité CDS²⁵

2. Dans la fenêtre suivante, ouvrir le dossier *Core Data Services* et sélectionner *Data Definition*. La création d'une *Data Definition* permet de développer une entité CDS. Notez que selon la version d'ADT, *Data Definition* se nomme *DDL source* ;

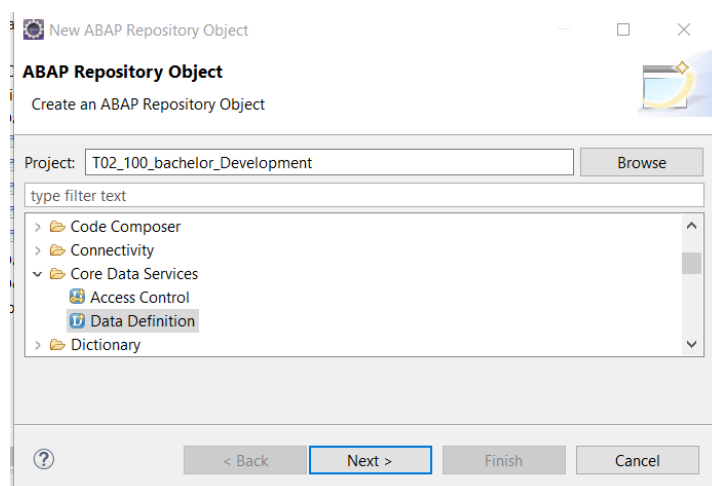


Figure 26 : Création entité CDS – ABAP Repository Object²⁶

²⁵ Source : Donnée de l'auteur sur Eclipse Neon

²⁶ Source : Donnée de l'auteur sur Eclipse Neon

3. Entrer les informations telles que le nom, en commençant par « Z », et une description ;

New Data Definition

Data Definition
Create a Data Definition

Project: * T02_100_bachelor_Development Browse...

Package: * ZP_BACHELOR_CDS Browse...

☐ Add to favorite packages

Name: * Z_CDS_Select

Description: * Select customers

Original language: EN

< Back Next > Finish Cancel

Figure 27 : Création entité CDS - Data Definition²⁷

4. Sélectionner un ordre de transport ;

New Data Definition

Selection of Transport Request

For Z_DD_CUSTOMER (Data Definition), the selected transport request will be used

☒ Choose from requests in which I am involved

Transport Req...	User	Target	Text
[REDACTED]	HES-SO Valais	Duc Marlene	BACHELOR

☐ Create a new request

Request description:

☐ Enter a request number

Request number: Browse...

< Back Next > Finish Cancel

Figure 28 : Création entité CDS - Transport Request²⁸

²⁷ Source : Donnée de l'auteur sur Eclipse Neon

²⁸ Source : Donnée de l'auteur sur Eclipse Neon

5. Sélectionner un *template*. Une sélection de modèles est à disposition pour créer une vue CDS avec l’affichage de la syntaxe par défaut. Les modèles existants sont les vues simples, avec jointure, avec association, avec paramètres, ainsi que les vues étendues.

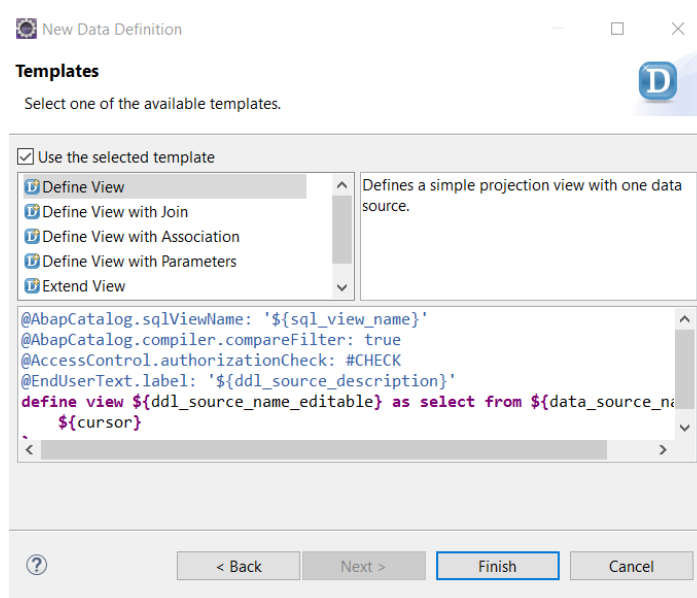


Figure 29 : Création entité CDS - Template²⁹

L'image ci-dessous présente les éléments du *template Define View* qu'il est possible de modifier.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_Customers'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_ALLOWED
4 @EndUserText.label: 'Select customers'
5
6 define view Z_CDS_Select as select from ztt_customer as customers{
7
8     key id_customer as Cutsomer_ID,
9     firstname as Firstname,
10    lastname as Lastname
11 }
12
13 where id_customer = 'DUM'

```

Figure 30 : Vue CDS³⁰



Les quatre points mis en évidence par un chiffre en rouge sont listés ci-après :

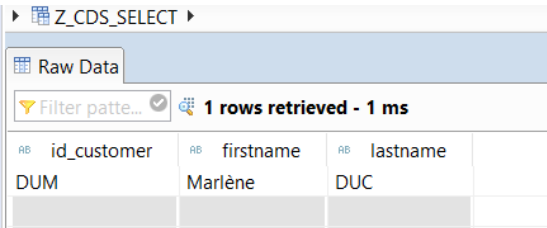
1. Les annotations obligatoires sont créées automatiquement avec la vue CDS (les autres annotations seront détaillées dans la section 5.4) ;

²⁹ Source : Donnée de l’auteur sur Eclipse Neon

³⁰ Source : Donnée de l’auteur sur Eclipse Neon

2. Les déclarations de la base de données :
 - a. Le type de la vue (*define view*) ;
 - b. Le nom de la vue CDS (*Z_CDS_Select*) ;
 - c. La source (*as select from*), suivie soit d'une table, soit d'une vue, avec éventuellement un alias (*as*) ;
 - d. L'ajout de jointures ou d'associations (détaillées dans les sections 5.5 et 5.6).
3. Les champs à sélectionner et les éventuelles fonctions et expressions. Il est possible d'utiliser le caractère « * » pour afficher tous les champs. Les alias peuvent également être ajoutés à chaque champ. Le mot clé *key*, inséré au début de la liste des champs, permet de définir l'élément comme clé de l'entité CDS ;
4. L'ajout d'éventuelles restrictions de sélection et d'instructions d'agrégation.

L'enregistrement de la vue CDS et son activation  sont maintenant possibles. La visualisation du résultat est alors disponible en exécutant la vue .



id_customer	firstname	lastname
DUM	Marlène	DUC

Figure 31 : Résultat de la requête CDS³¹

À la suite de l'activation, deux dossiers sont créés dans le package :

- *Core Data Services*, avec comme sous-dossier *Data Definitions* contenant les entités CDS ;
- *Views*, créé en tant que sous-dossier de *Dictionary*, où les vues SQL sont automatiquement créées.

En ouvrant une vue SQL, il est possible de naviguer à travers les différents onglets afin d'afficher par exemple les champs sélectionnés dans la source DDL, les éventuelles jointures ou les conditions de sélection. Il est important de noter que le message d'avertissement se trouvant au fond de la fenêtre indique que « les vues SQL ne sont supportées que de manière

³¹ Source : Donnée de l'auteur sur Eclipse Neon

limitée par SE11 ». Les modifications des vues SQL ne peuvent se faire qu'au travers de la vue CDS sous Eclipse.

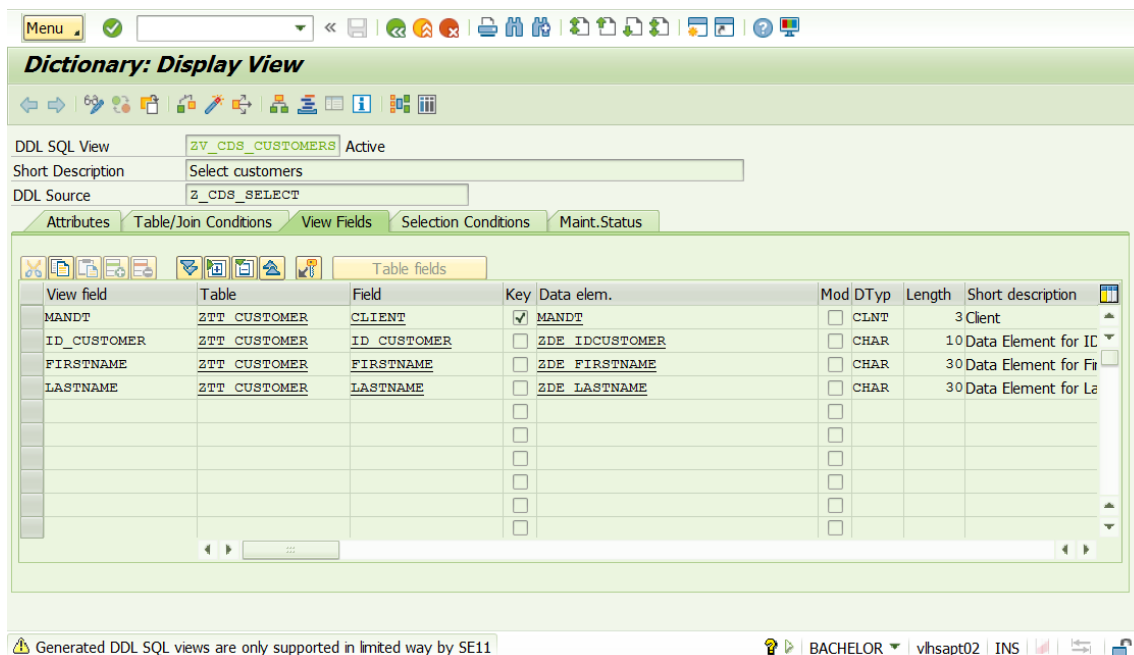


Figure 32 : Dictionary ABAP -Vue CDS³²

L'image ci-dessous résume les configurations possibles lors de la construction d'une vue CDS :

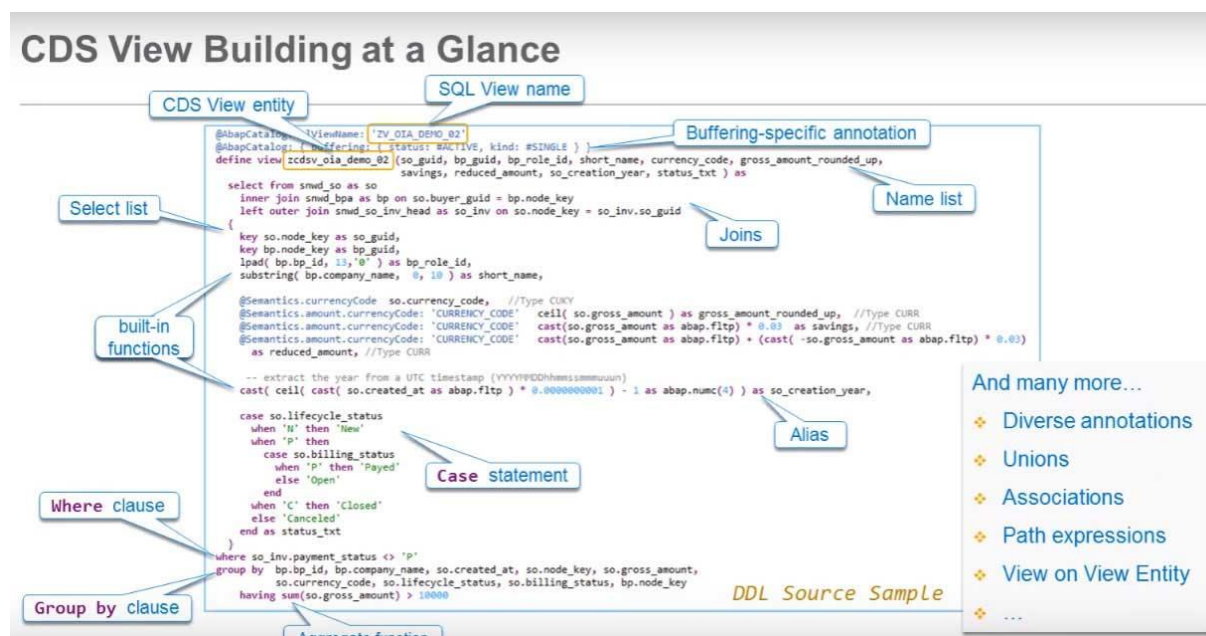


Figure 33 : Construction d'une vue CDS - Résumé³³

³² Source : Donnée de l'auteur sur SAP

³³ Source : <https://www.stechies.com/userfiles/CDS-source-view-2.jpg>

5.3. Utilisation d'une vue CDS dans un programme ABAP

L'utilisation d'une vue CDS dans un programme ABAP se fait de la même manière que pour une requête Open SQL avec comme source son nom.

```

1 *&-----
2 *& Report zprog_display_cds
3 *&-----
4 *&
5 *&-----
6 REPORT zprog_display_cds.
7
8 SELECT *
9     FROM z_cds_select
10    INTO TABLE @DATA(result).
11
12
13 cl_demo_output=>display_data(
14 EXPORTING
15     value = result
16     name = 'Display CDS View'
17 ).

```

Figure 34 : Programme ABAP avec vue CDS³⁴

5.4. Annotations

Pour cet exemple d'annotations, le choix s'est porté sur la table « ztt_material » afin de pouvoir utiliser une annotation de type `@Semantics` pour référencer les champs *currency* et *price*, en plus des annotations ajoutées automatiquement à la création.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_ANNOT' 1
2 @AbapCatalog.compiler.compareFilter: true 2
3 @AccessControl.authorizationCheck: #NOT_REQUIRED 3
4 @EndUserText.label: 'Annotations' 4
5
6 define view Z_CDS_ANNOTATION as select from ztt_material {
7
8     @EndUserText.label: 'Material ID' 5
9     key id_material,
10    @Semantics.currencyCode: true 6
11    currency,
12    @Semantics.amount.currencyCode: 'currency' 7
13    price
14
15 }

```

Figure 35 : Annotations CDS³⁵

³⁴ Donnée de l'auteur sur Eclipse Neon

³⁵ Source : Donnée de l'auteur sur Eclipse Neon

Les points mis en évidence par un chiffre en rouge sont explicités ci-après :

1. Cette annotation est obligatoire car elle permet de définir le nom de la vue SQL ;
2. Cette annotation définit si les conditions de filtrage doivent être comparées avec les expressions d'une vue. Les valeurs possibles sont :
 - a. *True* : La condition du filtre est comparée. Si elle correspond, la jointure associée est créée une seule fois. Les performances sont donc améliorées ;
 - b. *False* : Une jointure est créée séparément pour chaque condition de filtre.
3. Cette troisième annotation permet de spécifier si une autorisation doit être effectuée sur la vue CDS à travers un DCL. Les choix suivants sont disponibles :
 - a. *#NOT_ALLOWED* : Aucun contrôle n'est effectué ;
 - b. *#CHECK* : Un contrôle d'accès doit être effectué ;
 - c. *#NOT_REQUIRED* : Idem que pour la valeur *#CHECK*, à la différence que si aucun DCL n'est attribué à la vue CDS, il n'y a pas de message d'avertissement sur la syntaxe.
4. Cette annotation vise à décrire brièvement la vue CDS ;
5. Celle-ci donne une brève description de l'élément suivant ;
6. Cette annotation permet de définir l'élément suivant comme une clé de devise (*currency key*) ;
7. Enfin, la dernière annotation définit le champ lié avec la clé de devise précédemment établie.

5.5. Jointures

Pour la création d'une jointure, il est nécessaire de créer une nouvelle entité CDS en sélectionnant le *template Define view with join*.

Dans l'exemple ci-dessous, une jointure de type *JOIN* est utilisée pour lier la table « ztt_order » à « ztt_customer ». La condition *ON* permet de définir la condition pour lier les deux tables, soit à travers un nom commun, soit à travers des clés étrangères.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_JOIN'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK
4 @EndUserText.label: 'Join between tables'
5 define view Z_CDS_Join as select from ztt_order as orders
6 join ztt_customer as customers
7   on orders.id_customer = customers.id_customer {
8
9   key id_order,
10  orders.id_customer,
11  firstname,
12  lastname,
13  order_date
14 }

```

Figure 36 : Jointure CDS³⁶

5.6. Associations

Pour la création d'une association, il est nécessaire de créer une nouvelle entité CDS en sélectionnant le *template Define view with Association*.

Dans l'exemple ci-dessous, l'association est utilisée pour lier la table « ztt_customer » à « ztt_order ». Comme pour la jointure, la condition *ON* permet de définir la condition pour lier les deux tables. La cardinalité de l'association est définie après le mot clé *association* et son alias doit comporter le caractère « _ » devant le nom. Il est possible d'exposer l'association pour pouvoir la réutiliser dans une autre entité CDS, comme sous le point 1 de l'image.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_Assoc'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_ALLOWED
4 @EndUserText.label: 'Associations'
5 define view Z_CDS_Association as select from ztt_customer as customers
6 association [1..*] to ztt_order as _orders
7   on customers.id_customer = _orders.id_customer
8 {
9 key id_customer as Id_Customer,
10  firstname as Firstname,
11  lastname as Lastname,
12  /* Associations */
13  _orders.id_order,
14  _orders 1
15
16 }

```

Figure 37 : Association CDS³⁷

Dans l'exemple suivant, l'entité CDS « Z_CDS_Association » est utilisée comme référence, afin de pouvoir réutiliser l'association préalablement exposée et lui ajouter un filtre à travers

³⁶ Source : Donnée de l'auteur sur Eclipse Neon

³⁷ Source : Donnée de l'auteur sur Eclipse Neon

une cardinalité. Ci-dessous, le filtre permet d'afficher l'*id* de la commande correspondant à l'*id* du client DUM.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_ASS_REU'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_ALLOWED
4 @EndUserText.label: 'Reuse association and filter'
5 define view Z_CDS_ASSOCIATION_REUSE as select from Z_CDS_Association {
6
7     _orders[1: id_customer = 'DUM'].id_order as Id_Order
8
9 }

```

Figure 38 : Association CDS - Réutilisation et filtre³⁸

Les associations peuvent être récupérées dans un programme ABAP à travers la syntaxe suivante :

```

SELECT \_orders-order_date
      FROM z_cds_association
      INTO @DATA(result).
ENDSELECT.

```

Figure 39 : Association CDS - Programme ABAP³⁹

5.7. Autorisations

Pour limiter les résultats retournés par l'entité CDS selon les utilisateurs, il est possible de créer un contrôle d'accès, en relation avec un objet d'autorisation existant. Le détail sur l'objet d'autorisation utilisé dans cet exemple est disponible dans les annexes.

Sous Eclipse, la création d'une source DCL est effectuée en créant un nouveau contrôle d'accès, comme illustré par l'image ci-dessous.

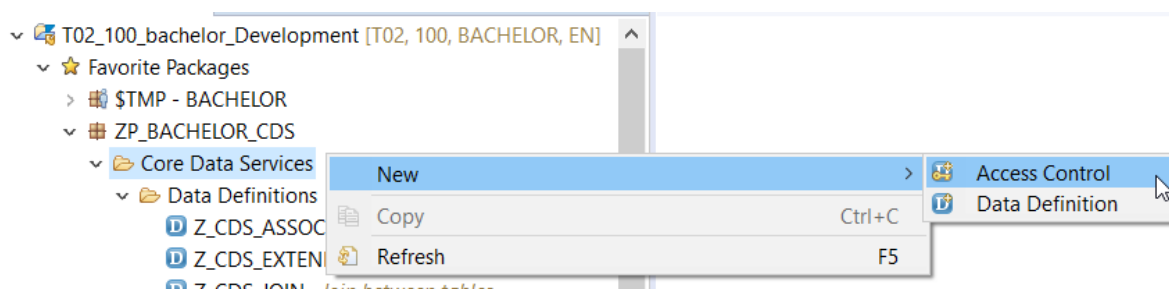


Figure 40 : Création d'un contrôle d'accès⁴⁰

Comme pour la création des entités CDS, un nom, une description et l'ordre de transport

³⁸ Source : Donnée de l'auteur sur Eclipse Neon

³⁹ Source : Donnée de l'auteur sur Eclipse Neon

⁴⁰ Source : Donnée de l'auteur sur Eclipse Neon

doivent être attribués. L'image ci-dessous affiche le *template* mis à disposition.

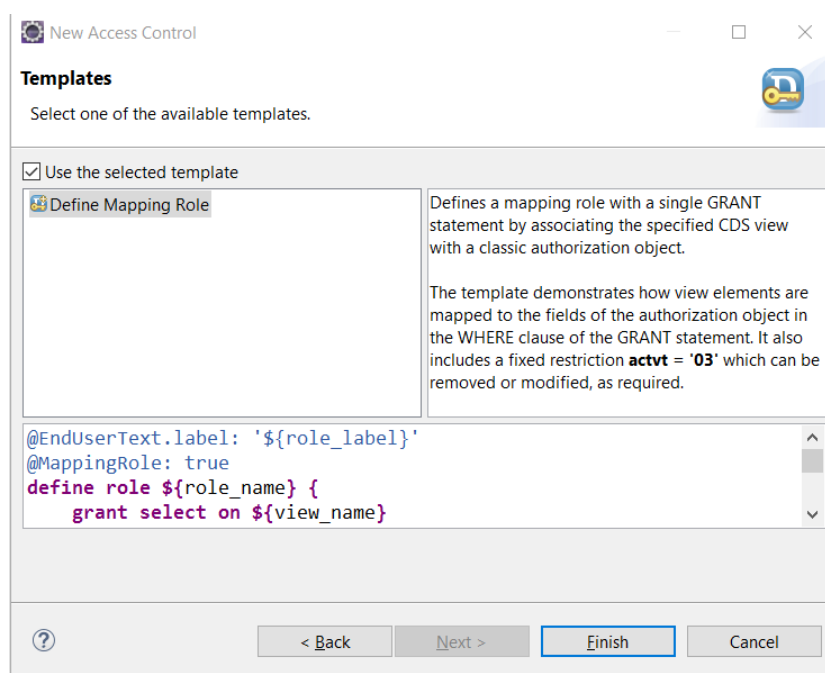


Figure 41 : Contrôle d'accès - Templates⁴¹

Ci-dessous est présenté un exemple d'autorisation avec, pour chaque point à définir (en rouge sur l'illustration), une explication.

```
1 @EndUserText.label: 'CUSTOMER Authorization' 1
2 @MappingRole: true
3 define role Z_CUSTOMER_ROLE 2{
4   grant select on Z_CDS_AUTHORIZ 3
5   where (id_customer) = 4
6   aspect pfcg_auth (ZAO_USER, CUSTOMER, actvt = '03' ) ; 5
7
8 }
```

Figure 42 : Contrôle d'accès CDS⁴²

1. Le label d'information, lequel doit être explicite ;
2. Le nom du rôle ;
3. Le nom de la vue CDS sur laquelle s'applique le contrôle des autorisations ;
4. Le champ sur lequel l'autorisation doit être vérifiée ;
5. Les détails relatifs à l'objet d'autorisation. Le premier champ entre parenthèses

⁴¹ Source : Donnée de l'auteur sur Eclipse Neon

⁴² Source : Donnée de l'auteur sur Eclipse Neon

correspond au nom de l'objet d'autorisation et les deux suivants aux champs définis dans l'objet d'autorisation.

Une fois le DCL activé, un sous-dossier *Access Controls* est créé sous le dossier *Core Data Services*. Pour l'utiliser, il suffit de modifier dans la vue CDS l'annotation relative aux autorisations en y ajoutant le mot clé *#CHECK*.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_AUTHORIZ'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK ←
4 @EndUserText.label: 'Authorization'
5 define view Z_CDS_AUTHORIZ as select from ztt_customer{
6
7     key id_customer,
8     firstname,
9     lastname
10
11 }
```

Figure 43 : Contrôle d'accès CDS - Autorisation⁴³

Sur la base du rôle simple défini sur SAP (voir annexe II), le résultat au moment de l'exécution de la vue CDS affiche uniquement le client Frédéric Morand.

Raw Data			
Filter patte...			
1 rows retrieved - 3 ms			
id_customer	firstname	lastname	
MOR	Frédéric	MORAND	

Figure 44 : Résultat avec le contrôle d'accès⁴⁴

Sans contrôle d'autorisation (*#NOT_ALLOWED*), le résultat affiche tous les clients.

Raw Data			
Filter patte...			
3 rows retrieved - 0 ms			
id_customer	firstname	lastname	
DUM	Marlène	DUC	
MAW	Werner	MAIER	
MOR	Frédéric	MORAND	

Figure 45 : Résultat sans le contrôle d'accès⁴⁵

⁴³ Source : Donnée de l'auteur sur Eclipse Neon

⁴⁴ Source : Donnée de l'auteur sur Eclipse Neon

⁴⁵ Source : Donnée de l'auteur sur Eclipse Neon

5.8. Extension de vues

La création d'une extension de vues est relativement simple. Comme pour les autres fonctionnalités, il faut créer une entité CDS en sélectionnant le *template Extend view*.

Dans ce *template*, à la suite du terme *extend view*, il faut indiquer la vue CDS à laquelle l'extension sera ajoutée. Dans l'exemple ci-dessous, la vue CDS de base est « Z_CDS_JOINS », laquelle joint les tables « ztt_order_details », « ztt_order » et « ztt_material ». L'extension permet de calculer le prix total.

```
1 @AbapCatalog.sqlViewAppendName: 'ZV_CDS_EXTVIEW'
2 @EndUserText.label: 'Extend view'
3 extend view Z_CDS_Joins with Z_CDS_EXTENDVIEW {
4
5 (quantity * price) as total_Price
6
7 }
```

Figure 46 : Extension CDS⁴⁶

Une fois l'extension activée, en analysant la vue CDS de base, une spirale apparaît, indiquant que l'entité possède des extensions de vue.

```
1 @AbapCatalog.sqlViewName: 'ZV_CDS_JOINS'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK
4 @EndUserText.label: 'Join multiple tables'
5 define view Z_CDS_Joins as select distinct from ztt_order_mater as details
6 join ztt_order as orders
```

Figure 47 : Signe d'une extension existante⁴⁷

En pointant la souris sur la spirale, une fenêtre affiche avec les informations suivantes : nom de la vue CDS et nom de l'extension.



Figure 48 : Affichage du nom de l'extension⁴⁸

A l'exécution de la vue CDS « Z_CDS_JOINS », le résultat affiché contient les champs de

⁴⁶ Source : Donnée de l'auteur sur Eclipse Neon

⁴⁷ Source : Donnée de l'auteur sur Eclipse Neon

⁴⁸ Source : Donnée de l'auteur sur Eclipse Neon

la vue de base avec les éléments de l'extension (*Total_Price*).

Raw Data									
Filter patte... 6 rows retrieved - 2 ms									
Customer_ID	Firstname	Lastname	Order_Date	Material_ID	Price	Quantity	Unit	Total_Price	
DUM	Marlène	DUC	2017-05-10	MDW	31.00	10	PC	310.00	
MAW	Werner	MAIER	2017-06-02	MDW	31.00	3	PC	93.00	
MOR	Frédéric	MORAND	2017-05-15	MDW	31.00	2	PC	62.00	
MAW	Werner	MAIER	2017-06-02	CLAYCL650	1350.00	1	PC	1350.00	
MOR	Frédéric	MORAND	2017-05-15	MB11/2C	58.00	1	PC	58.00	
MOR	Frédéric	MORAND	2017-05-15	TRBE111	586.00	1	PC	586.00	

Figure 49 : Résultat de l'extension de vues⁴⁹

5.9. Paramètres

Pour cet exemple utilisant les paramètres, toutes les tables sont liées, permettant ainsi d'afficher la totalité des informations concernant une commande, selon le nom et le prénom passés en paramètres.

```

1 @AbapCatalog.sqlViewName: `ZV_CDS_PARAMETER`
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_ALLOWED
4 @EndUserText.label: 'Parameter'
5 define view Z_CDS_Parameter
6     with parameters customer_Firstname : zde_firstname, 1
7                     customer_Lastname : zde_lastname
8 as select distinct from ztt_order_mater as details
9 join ztt_order as orders
10    on details.id_order = orders.id_order
11 join ztt_customer as customers
12    on orders.id_customer = customers.id_customer
13 join ztt_material as material
14    on details.id_material = material.id_material{
15    orders.id_customer as Customer_ID,
16    $parameters.customer_Firstname as Firstname,
17    $parameters.customer_Lastname as Lastname, 2
18    order_date as Order_Date,
19    details.id_material as Material_ID,
20    price as Price,
21    quantity as Quantity,
22    unit as Unit
23
24 } where customers.firstname = $parameters.customer_Firstname 3
25 and customers.lastname = $parameters.customer_Lastname

```

Figure 50 : Paramètres CDS⁵⁰

Les points mis en évidence par un chiffre en rouge sont explicités ci-après :

1. L'ajout de paramètres se fait avant la sélection des tables en leur assignant un nom et en spécifiant leur type. Dans l'exemple ci-dessus, les types choisis sont des *Data Elements* du DDIC ;

⁴⁹ Source : Donnée de l'auteur sur Eclipse Neon

⁵⁰ Source : Donnée de l'auteur sur Eclipse Neon

2. L'utilisation des paramètres à travers la syntaxe *\$parameters* suivi du nom du paramètre peut être utilisé comme champ à afficher ;
3. A travers la clause *where*, l'affichage est restreint en fonction des paramètres.

Au moment de l'exécution, une fenêtre s'affiche et demande d'insérer le nom et le prénom du client.

Figure 51 : Paramètres CDS - Insertion des valeurs⁵¹

Le résultat affiche uniquement les données du client passé en paramètres :

Customer_ID	Firstname	Lastname	Order_Date	Material_ID	Price	Quantity	Unit
DUM	Marlène	DUC	2017-05-10	MDW	31.00	10	PC
DUM	Marlène	DUC	2017-06-20	MDW	31.00	2	PC

Figure 52 : Paramètres CDS - Résultat⁵²

⁵¹ Source : Donnée de l'auteur sur Eclipse Neon

⁵² Source : Donnée de l'auteur sur Eclipse Neon

L'utilisation des paramètres dans un programme ABAP se fait de la manière suivante :

```

1 *&-----
2 *& Report zprog_display_cds
3 *&-----
4 *&
5 *&-----
6 REPORT zprog_display_cds.
7
8
9 PARAMETERS p_fname type zde_firstname.
10 PARAMETERS p_lname type zde_lastname.
11
12 SELECT *
13     FROM z_cds_parameter( customer_Firstname = @p_fname,
14                          customer_Lastname = @p_lname )
15     INTO TABLE @DATA(result).
16
17
18 cl_demo_output=>display_data(
19     EXPORTING
20         value = result
21         name = 'Display CDS View'
22 ).
23

```

Figure 53 : Paramètres CDS - Utilisation dans un programme ABAP⁵³

5.10. Gestion des langues

Avec l'utilisation de la syntaxe `$session.system_language` dans la clause *where*, il est possible d'afficher les résultats selon la langue de la session. Dans l'exemple ci-dessous, les tables « ztt_material » et « ztt_material_t » (ayant le nom des matériaux en français et en anglais) sont utilisées.

```

1 @AbapCatalog.sqlViewName: 'ZV_CDS_LANG'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #NOT_ALLOWED
4 @EndUserText.label: 'Display data based on lang'
5 define view Z_CDS_LANG as select from ztt_material
6 inner join ztt_material_t on ztt_material.id_material = ztt_material_t.id_material {
7
8     key ztt_material.id_material,
9     name,
10    currency,
11    price,
12    lang
13
14 }where lang = $session.system_language

```

Figure 54 : Gestion des langues CDS⁵⁴

⁵³ Source : Donnée de l'auteur sur Eclipse Neon

⁵⁴ Source : Donnée de l'auteur sur Eclipse Neon

6. Prototype HANA CDS

La réalisation du prototype est faite sur la version d'Eclipse Neon.3 Release (4.6.3) et la version SAP S/4HANA 1511 SP02. Comme la création des entités CDS peut être faite de la même manière que pour ABAP CDS sur S/4HANA, il est plus intéressant de présenter comment créer des tables *.hdbdd* en utilisant le CDS sur SAP HANA. Pour ce faire, un compte avec des accès restreints (se référer à l'image ci-dessous) a été créé pour HANA DB et HANA Studio.

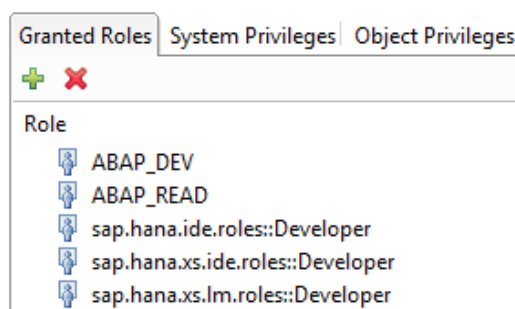


Figure 55 : Prototype HANA CDS - Droits de l'utilisateur⁵⁵

6.1. Configuration de l'environnement Eclipse

Cette partie permet de configurer l'environnement de développement Eclipse en y ajoutant les outils de développement et la connexion à la base de données.

6.1.1. Outils de développement HANA

L'installation du plug-in de développement HANA est identique à celui d'ABAP à l'exception des outils de développement à sélectionner, soit dans ce cas « *SAP Cloud Platform Tools* » et « *SAP HANA Tools* » (uiravikumar@gmail.com, 2017).

⁵⁵ Source : Donnée de l'auteur

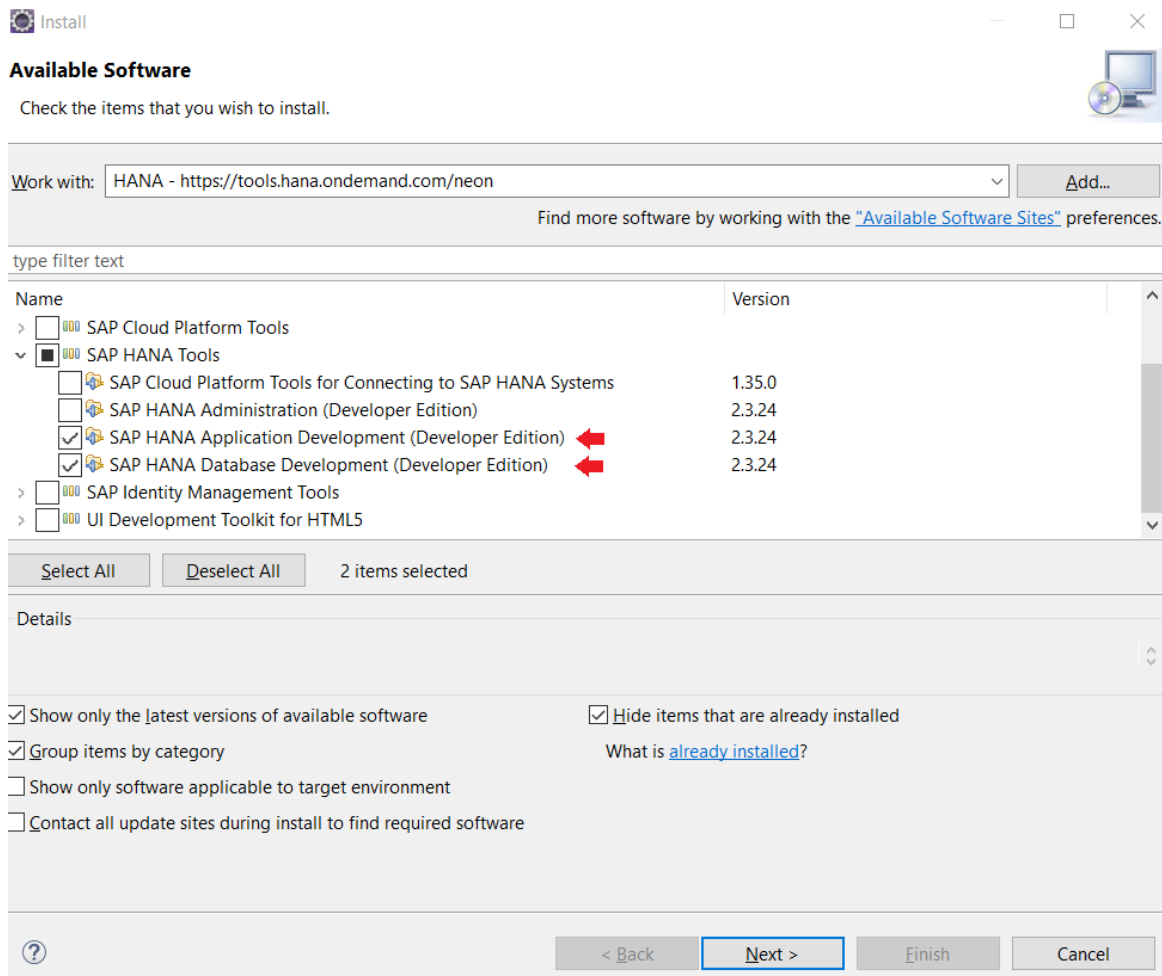


Figure 56 : Installation ADT Tools "Available Software" HANA⁵⁶

6.1.2. Ajouter un système SAP HANA à Eclipse

Les artefacts de développement d'applications sont stockés et gérés à travers SAP HANA repository. Afin de pouvoir s'y connecter, il faut ajouter un système dans Eclipse.

Avant de pouvoir ajouter un système, il faut changer de perspective pour SAP HANA, en suivant le chemin : *Window > Perspective > Open Perspective > Other > SAP HANA Development* (Portal, 2017).

⁵⁶ Source : Donnée de l'auteur sur Eclipse Neon

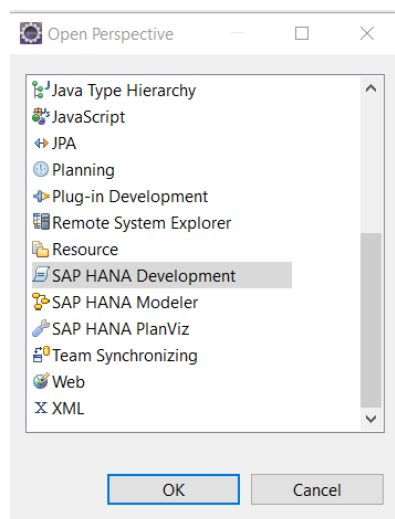


Figure 57 : Configuration Eclipse - Perspective HANA⁵⁷

L'ajout d'un nouveau système se fait via l'onglet « Systems ».

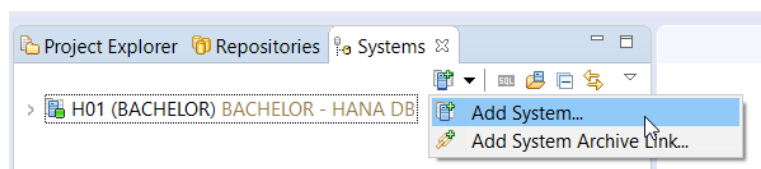


Figure 58 : Configuration Eclipse - Nouveau système HANA⁵⁸

Les informations relatives au système doivent être spécifiées.

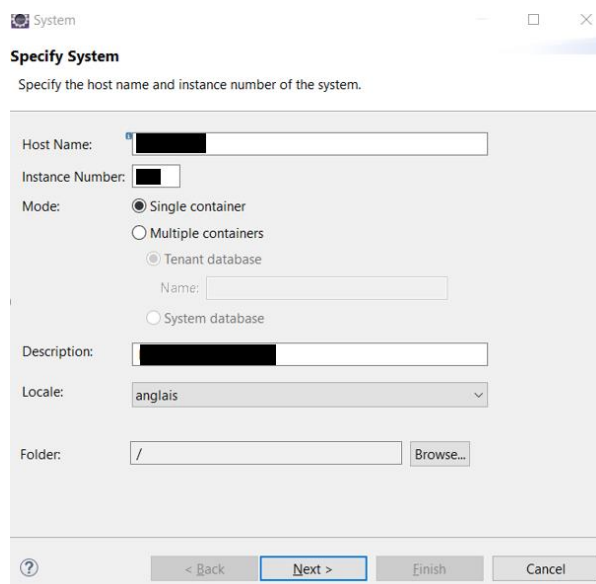


Figure 59 : Configuration Eclipse - Spécification du système HANA⁵⁹

⁵⁷ Source : Donnée de l'auteur sur Eclipse Neon

⁵⁸ Source : Donnée de l'auteur sur Eclipse Neon

⁵⁹ Source : Donnée de l'auteur sur Eclipse Neon

La dernière étape de la configuration consiste à spécifier les informations relatives à la connexion.

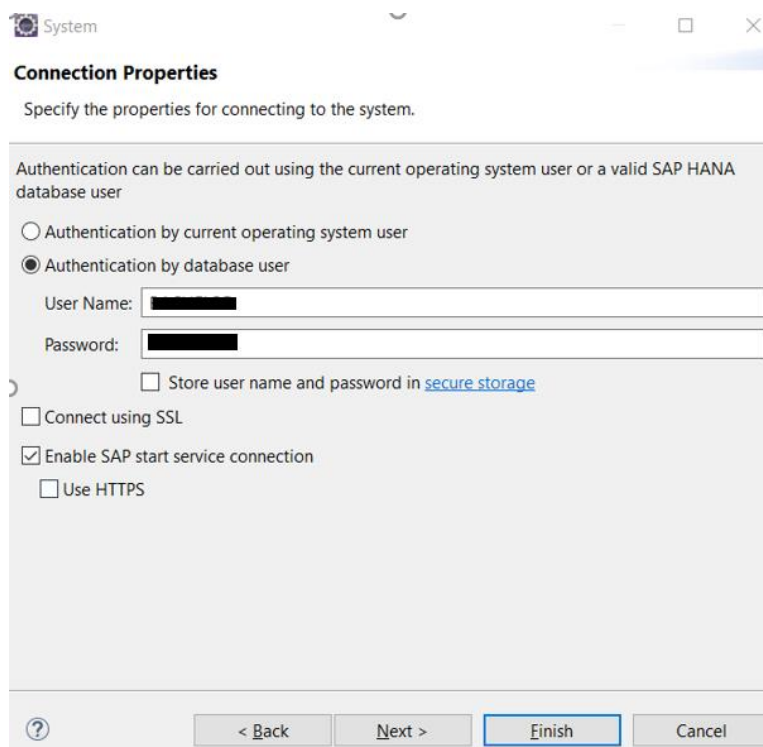


Figure 60 : Configuration Eclipse - Propriétés de connexion HANA⁶⁰

Suite à l'ajout du nouveau système, le Workspace est automatiquement ajouté sous l'onglet *Repositories*.

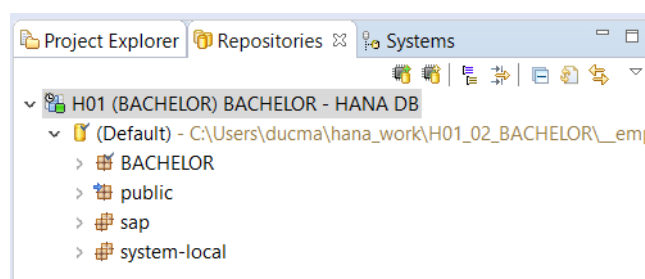


Figure 61 : Configuration Eclipse - Repositories Workspace HANA⁶¹

6.2. Création des tables

La description d'un modèle de persistance se fait à travers un document CDS. Celui-ci correspond à un fichier source contenant la définition des objets dont la création est souhaitée dans le catalogue SAP HANA (Portal, 2017).

⁶⁰ Source : Donnée de l'auteur sur Eclipse Neon

⁶¹ Source : Donnée de l'auteur sur Eclipse Neon

La création d'un document CDS se réalise de la manière suivante (Portal, 2017) :

1. Sous l'onglet *Repositories*, sélectionner le package sous lequel le document doit être créé en utilisant le clic droit et en suivant le chemin : *New > Other > Database Development > DDL Source File* ;

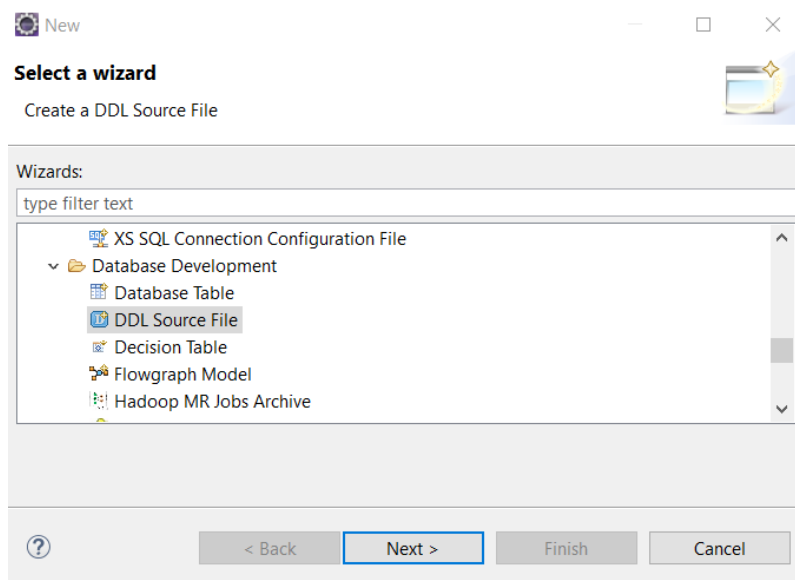


Figure 62 : Création d'un document CDS⁶²

2. Insérer le nom du document CDS.

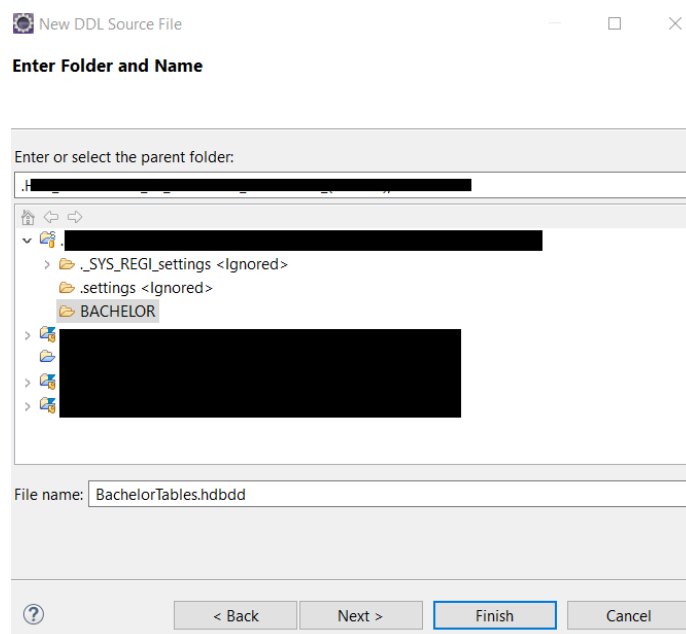


Figure 63 : Création d'un document CDS - Sélection fichier⁶³

⁶² Source : Donnée de l'auteur sur Eclipse Neon

⁶³ Source : Donnée de l'auteur sur Eclipse Neon

L'image ci-dessous présente les éléments à insérer dans le document CDS pour la création des tables.

```
namespace BACHELOR; 1

@Schema: 'BACHELOR' 2
context BachelorTables { 3

    type SString : String(20);
    type SDate : LocalDate; 4

    @Catalog.tableType: #COLUMN
    entity Customers{
        key id_customer : SString;
        Firstname : SString; 5
        Lastname : SString;
    };

    @Catalog.tableType: #COLUMN
    @nokey
    entity Orders{
        id_order : SString;
        Order_DATE : SDate;
        id_customer : association to Customers; 6
    };

};
```

Figure 64 : Document CDS⁶⁴

Les points mis en évidence par un chiffre en rouge sont explicités ci-après :

1. *Namespace* : le nom du *package repository* dans lequel le document CDS doit être créé ;
2. *Annotations* : les annotations disponibles pour le document CDS sont :
 - a. *@Catalog.index* : utilisé pour définir un index ;
 - b. *@Catalog.tableType* : permet de spécifier le type de la table, soit en colonne, soit en ligne, soit temporaire ;
 - c. *@nokey* : permet d'indiquer que l'entité ne possède pas de clé primaire ;
 - d. *@Schema* : indique le nom du schéma à utiliser pour stocker les entités ;
 - e. *@GenerateTableType* : lorsqu'une structure *type* est définie, *table type* est généré dans le catalogue. Il est possible d'utiliser une structure *type* sans générer de *table type* dans le catalogue en lui assignant la valeur *false* ;
 - f. *@SearchIndex* : utilisé pour définir quels éléments de la colonne doivent être indexés pour les fonctions de recherche ;
 - g. *@WithStructuredPrivilegeCheck* : permet de contrôler l'accès aux données (lors de la création d'une vue) selon les privilèges définis avec le DCL ;

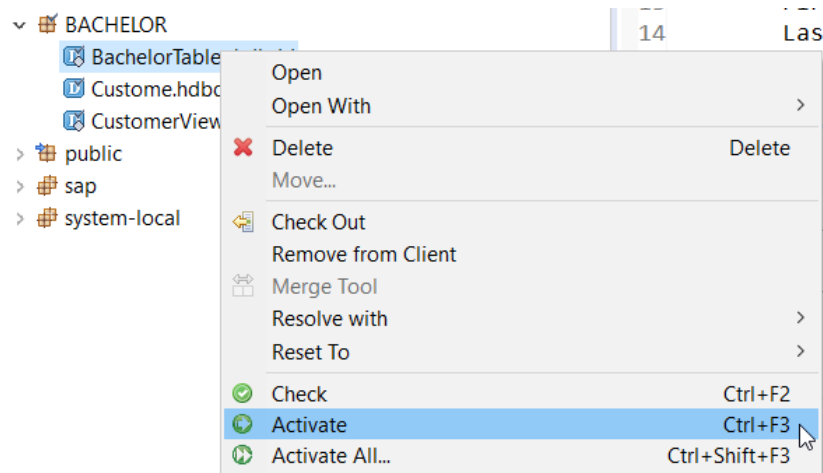
⁶⁴ Source : Donnée de l'auteur sur Eclipse Neon

3. *Context* : le contexte permet de grouper des artefacts. Un document CDS possède un seul contexte à la racine, mais dans celui-ci, il est possible d'en ajouter d'autres selon les besoins ;
4. *Type* : il est possible de définir des types dans le document CDS ;
5. *Entity* : l'entité est utilisée pour créer de nouvelles tables en y insérant le nom des champs et leurs types ;
6. *Associations* : elles permettent de définir les relations entre les entités. Si aucune cardinalité n'est précisée, par défaut, elle correspond à [0..1] et la clé étrangère est définie automatiquement en utilisant la clé primaire de l'entité associée. Il est également possible de spécifier les clés étrangères, comme illustré par la figure ci-après :

```
@Catalog.tableType: #COLUMN
@nokey
entity Orders{
  id_order : SString;
  Order_DATE : SDate;
  id_customer : Association to Customers {id_customer, Firstname};
};
```

Figure 65 : Documents CDS - Association⁶⁵

Une fois que le document est complet, il faut l'activer en choisissant *Activate*.

Figure 66 : Document CDS - Activation⁶⁶

⁶⁵ Source : Donnée de l'auteur sur Eclipse Neon

⁶⁶ Source : Donnée de l'auteur sur Eclipse Neon

Suite à l'activation, il est possible de constater sous l'onglet *Systems* la création des tables.

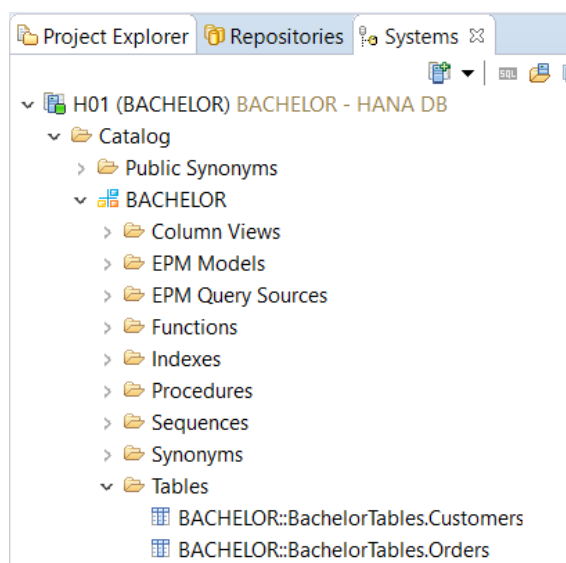


Figure 67 : BD HANA - Tables⁶⁷

⁶⁷ Source : Donnée de l'auteur sur Eclipse Neon

7. Avantages et inconvénients par rapport à l'ABAP Dictionary

Il est important de relever que le CDS a été inventé par SAP pour pallier l'incapacité d'*ABAP Dictionary* et de *SAP HANA Studio* de répondre aux besoins de toutes les applications métier (Ahmed, 2017). Evidemment, selon les besoins des entreprises, le CDS n'est pas toujours un avantage. Par exemple, si une vue n'est utilisée qu'une seule fois, la création de vues CDS n'est pas réellement nécessaire et n'apporte pas d'avantages par rapport à une vue normale. Cependant, si les besoins demandent de pouvoir réutiliser une vue ou demandent des capacités sémantiques ou techniques supérieures à celles offertes par Open SQL, le CDS est un avantage.

Parmi les fonctionnalités offertes et décrites dans le présent rapport, les annotations permettent un usage flexible dans des contextes différents (*OData*, *Business Logic*,...) et constituent par conséquent un apport intéressant du CDS. La possibilité de gérer les autorisations directement sur la vue CDS offre également l'avantage de ne plus avoir besoin de les gérer dans un programme ABAP. De plus, l'extension de vues permet non seulement d'optimiser un ensemble de résultats en transférant un minimum de données, mais également d'ajouter des colonnes supplémentaires, des expressions ou des associations à une vue existante.

SAP ne délaisse pas pour autant l'*ABAP Dictionary*, en proposant des améliorations d'Open SQL afin d'accroître les performances et de se rapprocher de l'Open SQL.

Cependant et comme démontré par Sujin Appukuttan dans son post du 9 septembre 2016, « ABAP new Open SQL and CDS runtime », les améliorations d'Open SQL présentent un temps d'exécution légèrement plus rapide que les vues CDS (résultat en microsecondes) (Appukuttan, 2017) :

Old Sql	CDS View	Enhanced Sql
66.95	49.47	46.83
64.07	62.51	52.07
12.65	17.46	10.02

Figure 68 : Temps d'exécution Open SQL VS CDS View⁶⁸

⁶⁸ Source : <https://blogs.sap.com/2016/09/09/abap-new-open-sql-and-cds/>

8. Conclusion

8.1. Synthèse générale

L'étude théorique du CDS et de ses fonctionnalités a permis de maîtriser son concept et les possibilités qu'il offre. La partie pratique a permis de vérifier l'exactitude des recherches, d'adapter la partie théorique et de mettre en pratique les différentes fonctionnalités du CDS à travers deux prototypes.

Dans le cadre de ce travail, il m'a été permis de connaître les nouveaux systèmes SAP tels que SAP HANA, S4/ HANA et surtout le CDS. Un large choix de documentation officielle, de blogs, de vidéos et de tutoriels est accessible. Cela demande toutefois beaucoup de lectures et un important tri d'informations pour en ressortir les plus pertinentes et les mieux adaptées aux besoins de la présente étude.

Suite aux diverses recherches et au développement des prototypes, on peut conclure que SAP CDS possède des avantages majeurs par rapport à *l'ABAP Dictionary*. D'une part, il propose des améliorations et de nouvelles fonctions et expressions, mais surtout, SAP CDS propose trois fonctionnalités particulièrement séduisantes : les annotations, la gestion des autorisations et les associations. Les annotations permettent par exemple de connecter des tables existantes à de nouvelles technologies telles que OData et UI5. De son côté, la gestion des autorisations permet de définir les autorisations directement sur une vue et non sur un programme. Enfin, les associations définissent les relations entre les tables à l'aide des cardinalités.

Le CDS s'intègre à la gestion du cycle de vie des produits SAP et aux différentes technologies utilisées par SAP, notamment SAP NetWeaver Gateway. Cependant, selon les besoins des entreprises, le CDS n'apportera pas de plus-value et il sera préférable de rester sur *ABAP Dictionary*.

8.2. Evolutions futures

Les évolutions futures de ce nouveau concept sont très prometteuses et devraient permettre de convaincre les entreprises. Un aspect pourrait toutefois les dissuader d'adopter cette nouvelle technologie : sa mise en place et son intégration qui nécessitent des ressources humaines et financières importantes. L'avenir du CDS dépend donc de la volonté des entreprises clientes de SAP d'utiliser ce nouveau concept et des appréciations positives ou négatives qu'elles feront après son intégration.

Références

- Admin. (2017, 06 13). *Don't Try Coding ABAP Core Data Services Without Reading This First*. Retrieved from Information Technology Partners, INC:
<http://www.itpsap.com/blog/2017/03/18/dont-try-coding-abap-core-data-services-without-reading-this-first/>
- Administrator, S. (2017, 05 08). *What is SAP HANA?* Retrieved from SAP:
<https://archive.sap.com/documents/docs/DOC-60338>
- Ahmed, M. (2017, 05 24). *ABAP on SAP HANA. Part IV. Core Data Services*. Retrieved from SAP YARD - SAP Technical tips, solutions and more: <http://www.sapyard.com/abap-on-sap-hana-part-iv/>
- Alyapyshev, M. (2017, 06 20). *ABAP CDS views with Authorization based on Access Control*. Retrieved from SAP: <https://blogs.sap.com/2017/02/27/abap-cds-views-with-authorization-based-on-access-control/>
- An Overview of SAP S/4 HANA*. (2017, 05 22). Retrieved from SlideShare:
<https://fr.slideshare.net/DebajitBanerjee/an-overview-of-sap-s4hana-64499419>
- Appukuttan, S. (2017, 07 14). *ABAP new Open SQL and CDS runtime*. Retrieved from SAP Community: <https://blogs.sap.com/2016/09/09/abap-new-open-sql-and-cds/>
- Bisht, S. (2017, 05 24). *Core Data Services [CDS] in SAP S/4 HANA*. Retrieved from SAP:
<https://blogs.sap.com/2016/09/26/core-data-services-cds-in-sap-s4-hana/>
- Bizien, D. (2017, 19 05). *Point de vue sur SAP*. Retrieved from Qu'est ce que HANA ? :
<http://pdvsap.blogspot.ch/2014/06/quest-ce-que-hana.html>
- Bodas, R. (2017, 05 29). *HANA CDS Graphical Editor – Introduction*. Retrieved from SAP:
<https://blogs.sap.com/2017/01/12/hana-cds-graphical-editor-introduction/>
- Casabianca, F. (2017, 06 08). *Introduction à Open Data Protocol et WCF Data Services*. Retrieved from Developpez.com: <http://badger.developpez.com/tutoriels/dotnet/odata-wcf-data-services/#LI-C>
- Christophe. (2017, 05 17). *ACID*. Retrieved from Base de données: <http://www.base-de-donnees.com/acid/>
- company, A. S. (2016). *SAP HANA: The In-Memory Platform for Digital Transformation*. SAP Solution Brief.

- D028018. (2017, 06 12). *ABAP Development: Create a CDS View*. Retrieved from SAP - Tutorial Catalog: <https://www.sap.com/developer/tutorials/abap-dev-adt-create-cds-view.html>
- Guru99. (2017, 05 08). *Open SQL & Native SQL in SAP ABAP*. Retrieved from Guru99: <http://www.guru99.com/native-open-sql.html>
- Guru99. (2017, 19 05). *SAP Business Suite*. Retrieved from Guru99: <http://www.guru99.com/sap-business-suite.html>
- Illumiti. (2017, 05 22). *SAP S/4 HANA*. Retrieved from Illumiti - Realize your vision: <http://illumiti.com/sap-software/sap-hana/>
- Jung, T. (2017, 05 24). *New Core Data Services Features in SAP HANA 1.0 SPS 10*. Retrieved from SAP: <https://blogs.sap.com/2015/07/01/new-core-data-services-features-in-sap-hana-10-sps-10/>
- Karamba. (2017, 05 08). *Karamba ! décisionnel et développement durable*. Retrieved from Karamba: <http://www.karamba.fr/?p=2783>
- Keller, H. (2017, 07 04). *ABAP News for 7.40, SP08 – ABAP Core Data Services (CDS)*. Retrieved from SAP: <https://blogs.sap.com/2014/10/10/abap-news-for-740-sp08-abap-core-data-services-cds/>
- Keller, H. (2017, 07 04). *ABAP News for Release 7.50 – Annotations in ABAP CDS*. Retrieved from SAP: <https://blogs.sap.com/2015/11/13/abap-news-for-release-750-annotations-in-abap-cds/>
- Keller, H. (2017, 05 04). *CDS – One Concept, Two Flavors*. Retrieved from SAP Community: <https://blogs.sap.com/2015/07/20/cds-one-model-two-flavors/>
- Kessler, K. (2017, 05 08). *Enhanced ABAP Development with Core Data Services (CDS)*. Retrieved from SAP insider: <http://sapinsider.wispubs.com/Assets/Articles/2015/October/SPI-enhanced-ABAP-development-with-Core-Data-Services>
- olgadolinskaja. (2017, 05 12). *Create an ABAP project in Eclipse*. Retrieved from SAP - Tutorial Catalog: <https://www.sap.com/developer/tutorials/abap-create-project.html>
- Portal, S. H. (2017, 07 10). *SAP HANA Developer Guide for SAP HANA Studio*. Retrieved from SAP HANA Developer Guide: <https://help.sap.com/viewer/52715f71adba4aaeb480d946c742d1f6/2.0.01/en-US>
- Prasad, S. (2017, 06 20). *Access Control(DCL) in CDS View*. Retrieved from SAP Codes: <https://sapcodes.com/2016/10/20/access-controldcl-in-cds-view/>

Qu'est-ce que SAP NetWeaver. (2017, 05 05). Retrieved from Connaissances Informatique:

<http://www.ordinateur.cc/Logiciel/Logiciels-de-productivite%C3%A9/164583.html>

rédaction, L. (2017, 05 08). *Nouvelles fonctionnalités SAP EHP 7 pour SAP ERP 6.0* . Retrieved from

SAP: <http://news.sap.com/france/2016/03/02/nouvelles-fonctionnalites-sap-ehp-7-pour-sap-erp-6-0/>

Reddy, P. (2017, 05 08). *Core Data Services in ABAP*. Retrieved from SAP Community:

<https://blogs.sap.com/2016/02/22/core-data-services-in-abap/>

Rousse, M. (2017, 05 22). *SAP S/4HANA*. Retrieved from SearchSAP:

<http://searchsap.techtarget.com/definition/SAP-S-4HANA>

Santhosh, R. (2017, 06 16). *CDS Associations and Path Expressions – ABAP on HANA*. Retrieved

from SAP - Community Blog: <https://blogs.sap.com/2017/03/07/inner-join-with-cds-associations-abap-on-hana/>

SAP. (2014). *TERP 10 - SAP ERP : Intégration des processus de gestion*. Studio SAP.

SAP. (2017, 05 03). *ABAP - Keyword Documentation*. Retrieved from Help legacy:

https://help.sap.com/http.svc/rc/abapdocu_751_index_hm/7.51/en-US/abenabap.htm

SAP. (2017, 07 03). *ABAP CDS - Annotations* . Retrieved from ABAP - Keyword Documentation :

https://help.sap.com/http.svc/rc/abapdocu_751_index_hm/7.51/en-US/abencds_annotations.htm

SAP. (2017, 05 04). *ABAP Dictionary*. Retrieved from SAP Documentation:

https://help.sap.com/saphelp_nw73ehp1/helpdata/en/cf/21ea0b446011d189700000e8322d00/frameset.htm

SAP. (2017, 05 17). *Accédez à vos données en temps réel avec les services de base de données In-*

Memory de SAP HANA. Retrieved from SAP HANA:

<https://www.sap.com/suisse/products/hana/features/in-memory-database.html>

SAP. (2017, 05 08). *La suite logicielle ERP nouvelle génération* . Retrieved from SAP:

<https://www.sap.com/suisse/products/s4hana-erp.html>

SAP. (2017, 05 08). *Qu'est-ce que SAP HANA ?* Retrieved from SAP:

<https://www.sap.com/suisse/products/hana.html>

SAP. (2017, 05 22). *SAP - ABAP CDS Development User Guide*. Retrieved from SAP Documentation:

<https://help.sap.com/viewer/f2e545608079437ab165c105649b89db/7.5.6/en->

US/4ed1f2e06e391014adc9ffe4e204223.html

SAP. (2017, 05 08). *SAP Business Warehouse*. Retrieved from SAP:

<https://www.sap.com/suisse/products/business-warehouse.html>

SAP. (2017, 05 02). *SAP netweaver application server for ABAP 7.5*. Retrieved from SAP:

<https://www.sap.com/documents/2016/02/782ac129-5f7c-0010-82c7-eda71af511fa.html>

sapapprentice. (2017, 05 08). *Qu'est-ce que SAP Netweaver?* . Retrieved from Apprendre SAP /

Learn SAP: <https://sapapprentice.wordpress.com/2009/10/20/quest-ce-que-sap-netweaver/>

shyamuthaman. (2017, 07 19). *Insert data into custom tables – Flat file loads to SAP HANA*. Retrieved

from TeachMeHANA: <http://teachmehana.com/loading-data-flat-file-sap-hana/>

shyamuthaman. (2017, 07 19). *SAP HANA SQL Script INSERT*. Retrieved from TeachMeHANA:

<http://teachmehana.com/insert-statement-sql-script-sap-hana/>

Swanepoel, C. E. (2017, 07 04). *New ABAP CDS Tools in SAP NetWeaver AS ABAP 7.5*. Retrieved

from SAP Community: <https://blogs.sap.com/2015/10/22/new-abap-core-data-services-tools-in-sap-netweaver-as-abap-75/>

Swanepoel, C. E. (2017, 05 04). *New Data Modeling Features in SAP NW ABAP 7.4 SP5*. Retrieved

from SAP Community: <https://blogs.sap.com/2014/02/04/new-data-modeling-features-in-abap-for-hana/>

TECHED, S. (2017, 05 02). *DEV200 - Core Data Services - Next-Generation Data Definition and*

Access on SAP HANA. Retrieved from SAP:

<https://assets.cdn.sap.com/sapcom/docs/2016/02/bedb9729-5f7c-0010-82c7-eda71af511fa.pdf>

Technology, S. (2017, 05 08). *SAP HANA Platform – Overview* . Retrieved from SlidShare:

<https://fr.slideshare.net/SAPTechnology/sap-hana-platform-overview>

uiravikumar@gmail.com. (2017, 05 29). *How to Install SAP HANA Tools in Eclipse or HANA Studio*.

Retrieved from SAPLearners.com: <http://www.saplearners.com/how-to-install-sap-hana-tools-in-eclipse-or-hana-studio/>

Annexe I : Tables de la base de données

Les différentes tables créées et utilisées pour le prototype ABAP CDS, ainsi que leur contenu, sont disponibles ci-après.

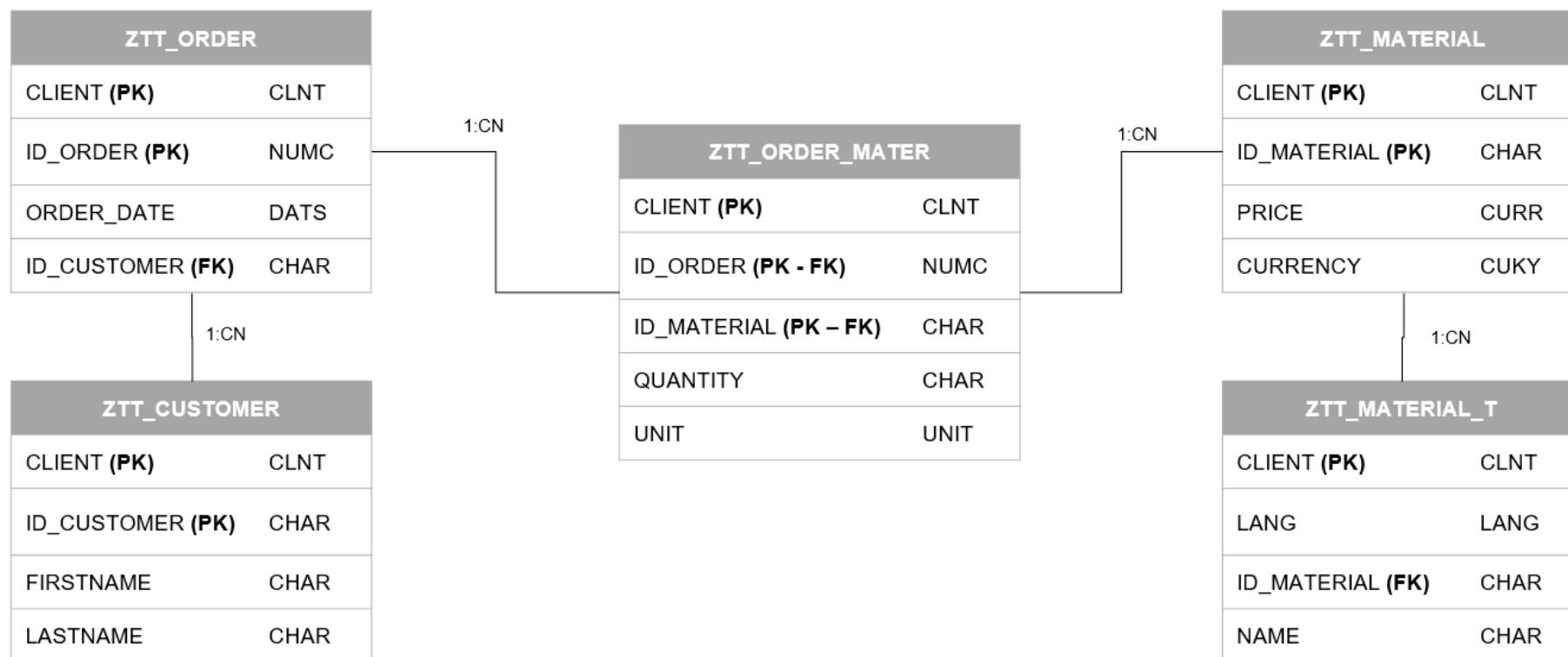


Figure 69 : Annexe I - Tables de base de données⁶⁹

⁶⁹ Source : Donnée de l'auteur

ZTT_ORDER			ZTT_CUSTOMER			ZTT_MATERIAL		
ID_ORDER	ORDER_DATE	ID_CUSTOMER	ID_CUSTOMER	FIRSTNAME	LASTNAME	ID_MATERIAL	PRICE	CURRENCY
1	10.05.2017	DUM	DUD	Damien	DUC	CLAYCL650	1 350,00	CHF
2	15.05.2017	MOR	DUM	Marlène	DUC	MB11/2C	58,00	CHF
3	02.06.2017	MAW	MAW	Werner	MAIER	MDW	31,00	CHF
4	20.06.2017	DUM	MOR	Frédéric	MORAND	TRBE111	586,00	CHF

ZTT_ORDER_MATER				ZTT_MATERIAL_T		
ID_ORDER	ID_MATERIAL	QUANTITY	UNIT	LANG	ID_MATERIAL	NAME
1	MDW	10	PC	EN	CLAYCL650	Clarinet Yamaha YCL-650
2	MB11/2C	1	PC	EN	MB11/2C	Mouth Bach 1-1/2C
2	MDW	2	PC	EN	MDW	Mute Denis Wick
2	TRBE111	1	PC	EN	TRBE111	Trumpet Besson BE111
3	CLAYCL650	1	PC	FR	CLAYCL650	Clarinete Yamaha YCL-650
3	MDW	3	PC	FR	MB11/2C	Embouchue Bach 1-1/2C
4	MDW	2	PC	FR	MDW	Sourdine Denis Wick
				FR	TRBE111	Trompette Besson BE111

Figure 70 : Annexe I - Contenu des tables de la base de données⁷⁰⁷⁰ Source : Donnée de l'auteur

Annexe II : Objet d'autorisation

Ci-dessous est présenté le détail des éléments créés pour les autorisations du prototype ABAP CDS (Prasad, 2017) (Alyapyshev, 2017) :

The screenshot shows two SAP windows. The main window is 'Display authorization object' with the following fields:

Object	ZAO_USER
Text	Authorization Object - User
Class	ZCDS CDS Authorization
Author	BACHELOR

Below these fields is a table for 'Authorization fields':

Authorization Field	Short Description
ACTVT	Activity
CUSTOMER	Data Element for ID Customer

At the bottom of the main window are two buttons: 'Display Object Documentation' and 'Permitted Activities'.

The 'Define Values' window is open on the right, showing the 'Activities' table:

S..	A...	Text
<input checked="" type="checkbox"/>	01	Add or Create
<input checked="" type="checkbox"/>	02	Change
<input checked="" type="checkbox"/>	03	Display

Figure 71 : Annexe II - Classe objet⁷¹

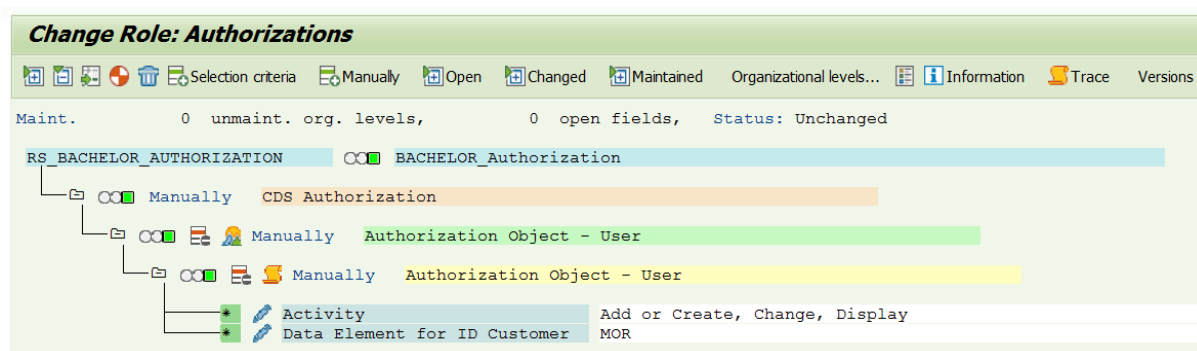


Figure 72 : Annexe II - Rôle simple et autorisations⁷²

⁷¹ Source : Donnée de l'auteur sur SAP

⁷² Source : Donnée de l'auteur sur SAP

Annexe III : Aides et astuces de développement

Ci-dessous sont présentées quelques aides et astuces disponibles sous Eclipse (Swanepoel, New ABAP CDS Tools in SAP NetWeaver AS ABAP 7.5, 2017) (SAP, SAP - ABAP CDS Development User Guide, 2017).

Code completion

En utilisant **CTRL + ESPACE** dans la zone prévue pour la sélection des champs, il est possible d'insérer tous les éléments de la base de données en un clic. Cette aide est plus utile pour les vues avec jointures ou associations qui n'autorisent pas la sélection de tous les éléments à travers l'utilisation du caractère « * ».

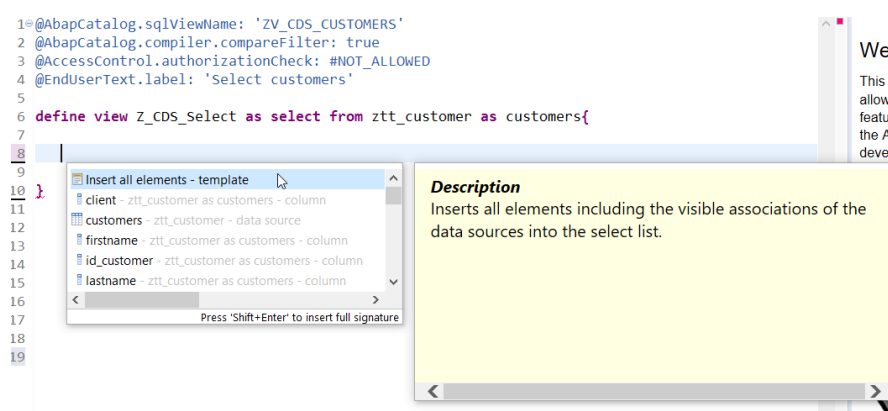


Figure 73 : Annexe III - Code completion⁷³

Wizard pour les clauses ON

Pour définir les clauses **ON** des jointures et des associations, il existe un *wizard*. Pour l'utiliser, il faut placer le curseur de la souris à la suite du mot clé **ON** et utiliser les touches du clavier **CTRL+1**.

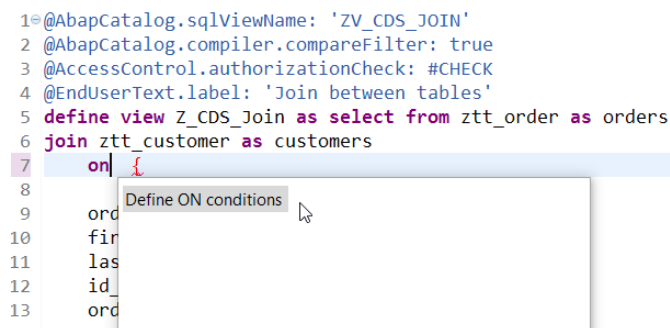


Figure 74 : Annexe III - Wizard pour les clauses « ON »⁷⁴

⁷³ Source : Donnée de l'auteur sur Eclipse Neon

⁷⁴ Source : Donnée de l'auteur sur Eclipse Neon

Il est alors possible de définir les conditions soit en utilisant les noms, soit en utilisant les clés étrangères, soit en utilisant l'action glisser/déposer un élément de la première table vers la deuxième.

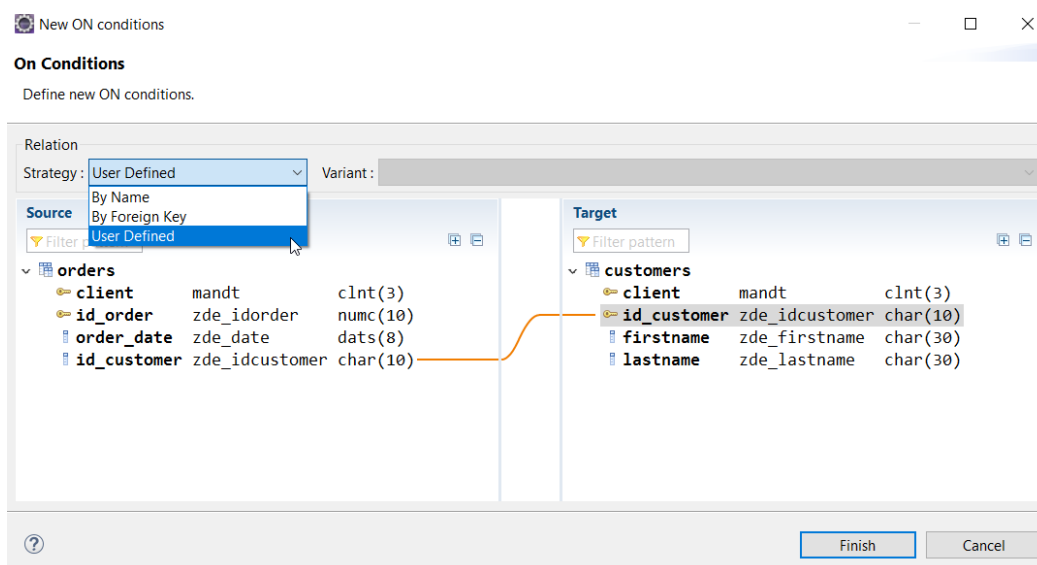


Figure 75 : Annexe III - Définition de la clause « ON »⁷⁵

Editeur Graphique

Il est possible d'afficher un éditeur graphique pour visualiser la source DDL et les relations existantes en suivant le chemin : *clic droit de la souris sur une source DDL > Open With > Graphical Editor*.

Ci-dessous se trouve un exemple affichant que la vue CDS « Z_CDS_JOIN » contient des annotations (@) et des restrictions (🔒). Les autres éléments sont grisés (🔒), ce qui indique qu'ils ne sont pas utilisés. En observant les liens, on constate que cette vue utilise les données venant de la table « ztt_order » et qu'il existe une jointure (🔗) vers la table « ztt_customer ».

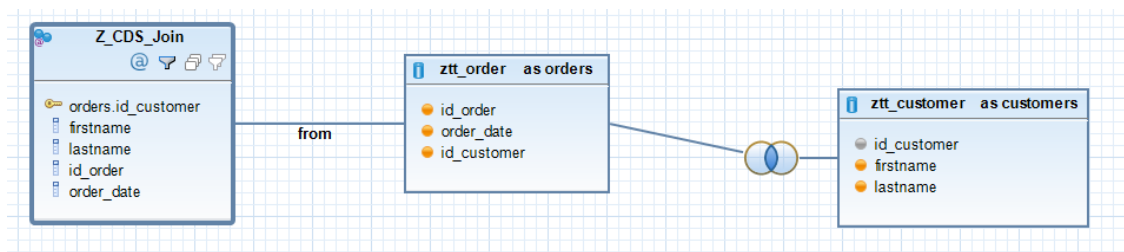


Figure 76 : Annexe III - Editeur Graphique⁷⁶

⁷⁵ Source : Donnée de l'auteur sur Eclipse Neon

⁷⁶ Source : Donnée de l'auteur sur Eclipse Neon

Active Annotations Views

Cet affichage permet de lister tous les éléments utilisés dans une vue CDS en suivant le chemin : *clic droit de la souris sur une source DDL > Open With > Active Annotations*.

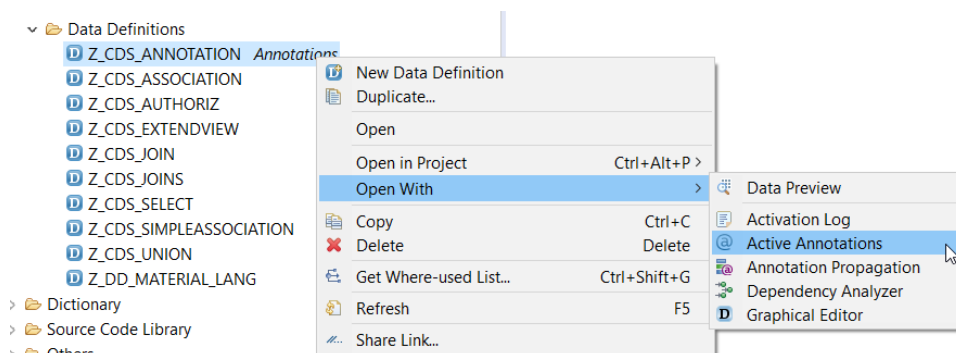


Figure 77 : Annexe III - Active Annotations Views⁷⁷

L'image ci-dessous résume les informations affichées grâce à *Active Annotations Views*.

Annotated Elements	Annotation Value	Origin Datasource	Origin Data Element
P_DisplayCurrency	Parameter		
ISO Currency Code	Field	SEPM_I_SalesOrderItemCube	SNWD_CURR_CODE
EPH: Sales Order Number		SEPM_I_SalesOrderItemCube	SNWD_SO_ID
Sales Order ID		SNWD_SO	SNWD_SO_ID
Sales Order ID		SNWD_SO	SNWD_SO_ID
Item Position		SEPM_I_SalesOrderItemCube	SNWD_SO_ITEM_POS
Sales Order Item Position		SNWD_SO_I	SNWD_SO_ITEM_POS
Sales Order Item Position		SNWD_SO_I	SNWD_SO_ITEM_POS

Figure 78 : Annexe III - Active Annotations Views Details⁷⁸

⁷⁷ Source : Donnée de l'auteur sur Eclipse Neon

⁷⁸ Source : <https://help.sap.com/viewer/f2e545608079437ab165c105649b89db/7.5.6/en-US/45ba68d3b4924a97ad2dc598e756968e.html>

Annexe IV : Insertion de données dans les tables HANA

L'insertion de données dans les tables HANA peut se faire soit à l'aide d'un script SQL soit à l'aide d'un fichier CSV.

Script SQL (shyamuthaman, SAP HANA SQL Script INSERT, 2017)

Depuis l'onglet *Systems*, il faut sélectionner l'icône SQL pour ouvrir la console SQL.

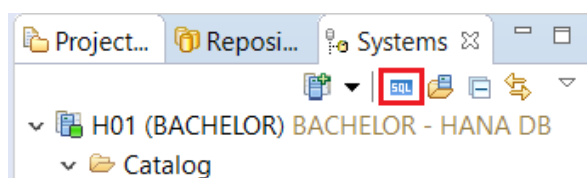


Figure 79 : Annexe IV - Ajout d'un nouveau Script SQL⁷⁹

Dans la console SQL, l'insertion des données se fait à l'aide de la requête *INSERT* suivante : **INSERT into tables values (value1, value2, value3);**.

```
SQL
1 INSERT into "BACHELOR"."BACHELOR::BachelorTables.Customers" values ('DUM','Marlene','DUC');
2 INSERT into "BACHELOR"."BACHELOR::BachelorTables.Customers" values ('DUD','Damien','DUC');
3 INSERT into "BACHELOR"."BACHELOR::BachelorTables.Customers" values ('MAW','Werner','MAIER');
4 INSERT into "BACHELOR"."BACHELOR::BachelorTables.Customers" values ('MOR','Frederic','Morand');
```

Figure 80 : Annexe IV - Script SQL⁸⁰

Il est alors possible d'afficher les données de la table en utilisant *Open Data Preview* et de vérifier la bonne insertion des données.

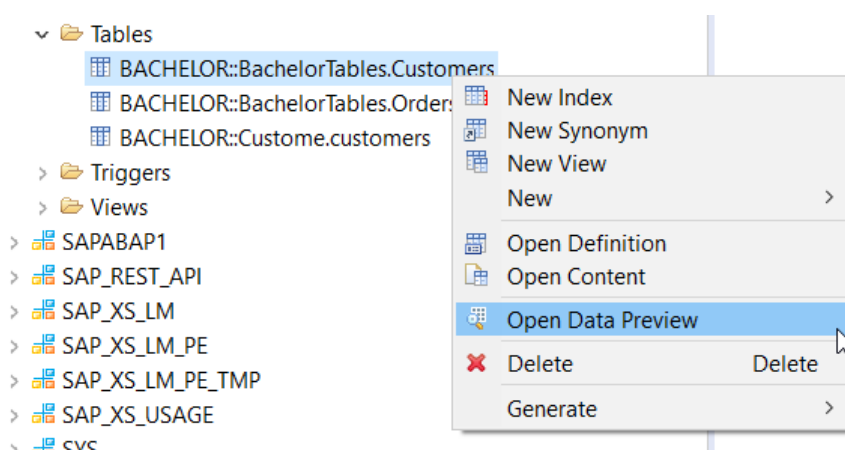
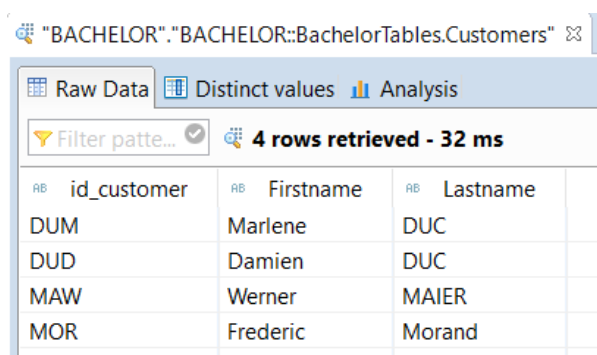


Figure 81 : Annexe IV - Open Data Preview⁸¹

⁷⁹ Source : Donnée de l'auteur sur Eclipse Neon

⁸⁰ Source : Donnée de l'auteur sur Eclipse Neon

⁸¹ Source : Donnée de l'auteur sur Eclipse Neon

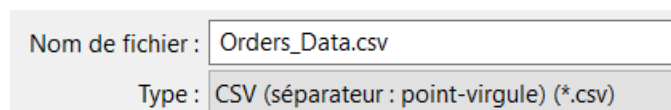


id_customer	Firstname	Lastname
DUM	Marlene	DUC
DUD	Damien	DUC
MAW	Werner	MAIER
MOR	Frederic	Morand

Figure 82 : Annexe IV – Données de la table⁸²

Fichier CSV (shyamuthaman, Insert data into custom tables – Flat file loads to SAP HANA, 2017)

Pour pouvoir importer un fichier csv, il est important que le format de son enregistrement soit correct : « CSV (séparateur : point-virgule)(*.csv) ».

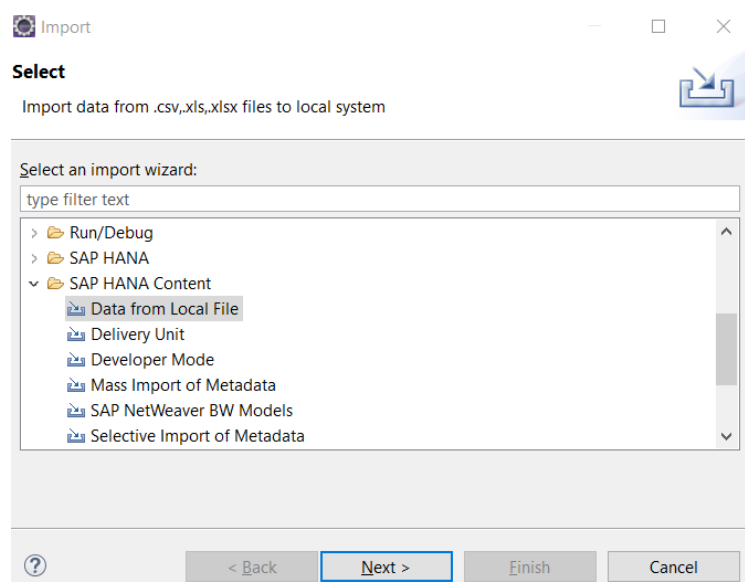


Nom de fichier : Orders_Data.csv

Type : CSV (séparateur : point-virgule) (*.csv)

Figure 83 : Annexe IV - Format du fichier csv⁸³

L'importation du fichier csv se fait sous Eclipse en utilisant le chemin suivant : *File > Import... > SAP HANA Content > Data from Local File*.

Figure 84 : Annexe IV - Import Data from Local File⁸⁴

⁸² Source : Donnée de l'auteur sur Eclipse Neon

⁸³ Source : Donnée de l'auteur

⁸⁴ Source : Donnée de l'auteur sur Eclipse Neon

L'écran suivant permet de sélectionner le système dans lequel le fichier csv doit être importé.

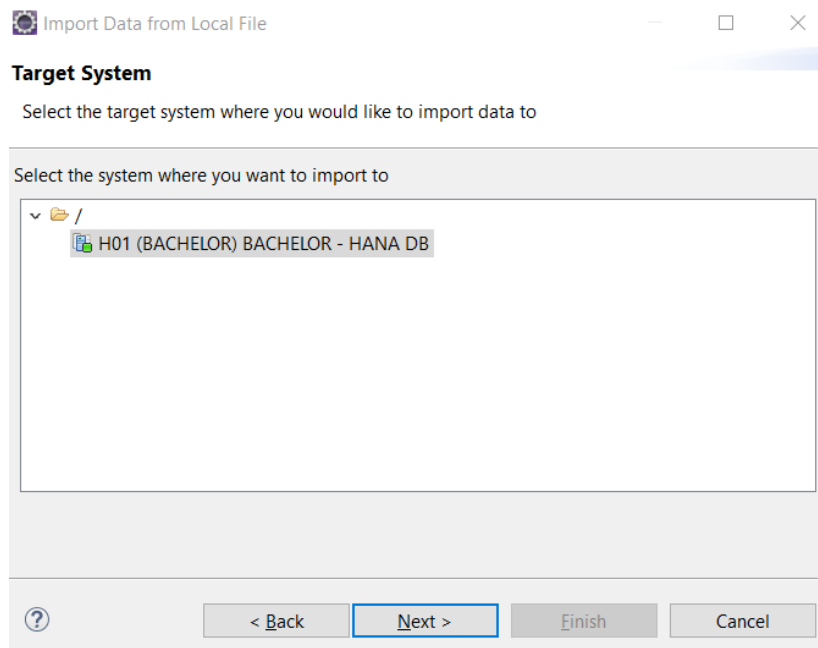


Figure 85 : Annexe IV - Import Data from Local File - Target System⁸⁵

L'écran suivant permet de définir les propriétés pour l'importation.

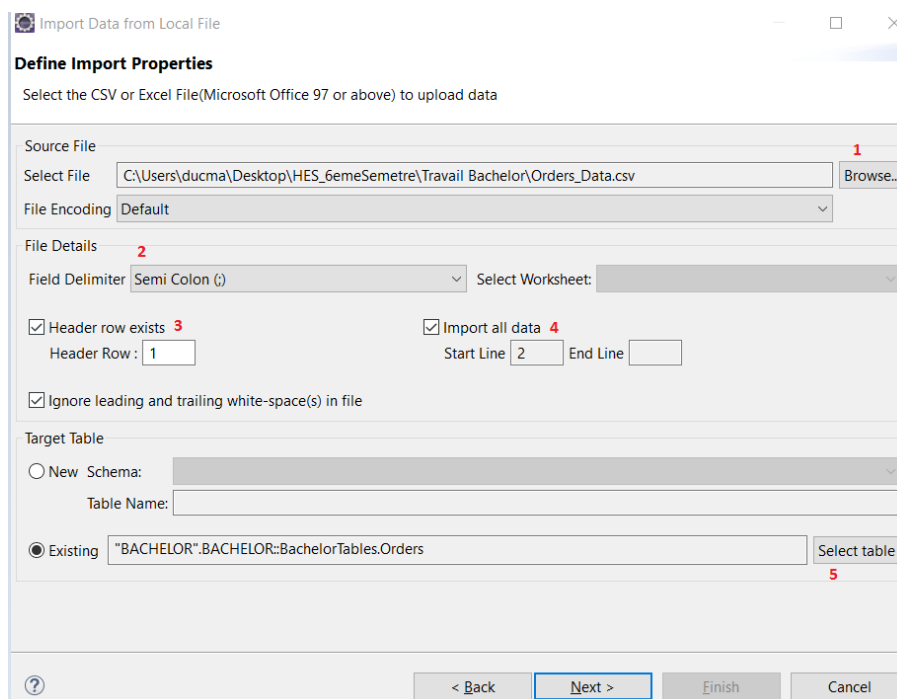


Figure 86 : Annexe IV - Import Data from Local File - Define Import Properties⁸⁶

⁸⁵ Source : Donnée de l'auteur sur Eclipse Neon

⁸⁶ Source : Donnée de l'auteur sur Eclipse Neon

Les points mis en évidence par un chiffre en rouge sont explicités ci-après :

1. Insérer le chemin du fichier csv à importer ;
2. Définir le type de séparateur utilisé dans le fichier csv ;
3. Si un en-tête existe dans le fichier, cocher la case et indiquer sa position ;
4. En cochant la case, toutes les données sont importées. Sinon, il est possible d'indiquer les lignes à importer ;
5. Sélectionner la table où les données doivent être importées.

L'écran suivant permet de mapper les données entre le fichier csv et la table. Le mappage se fait en glissant-déposant les champs du fichier csv vers ceux de la table.

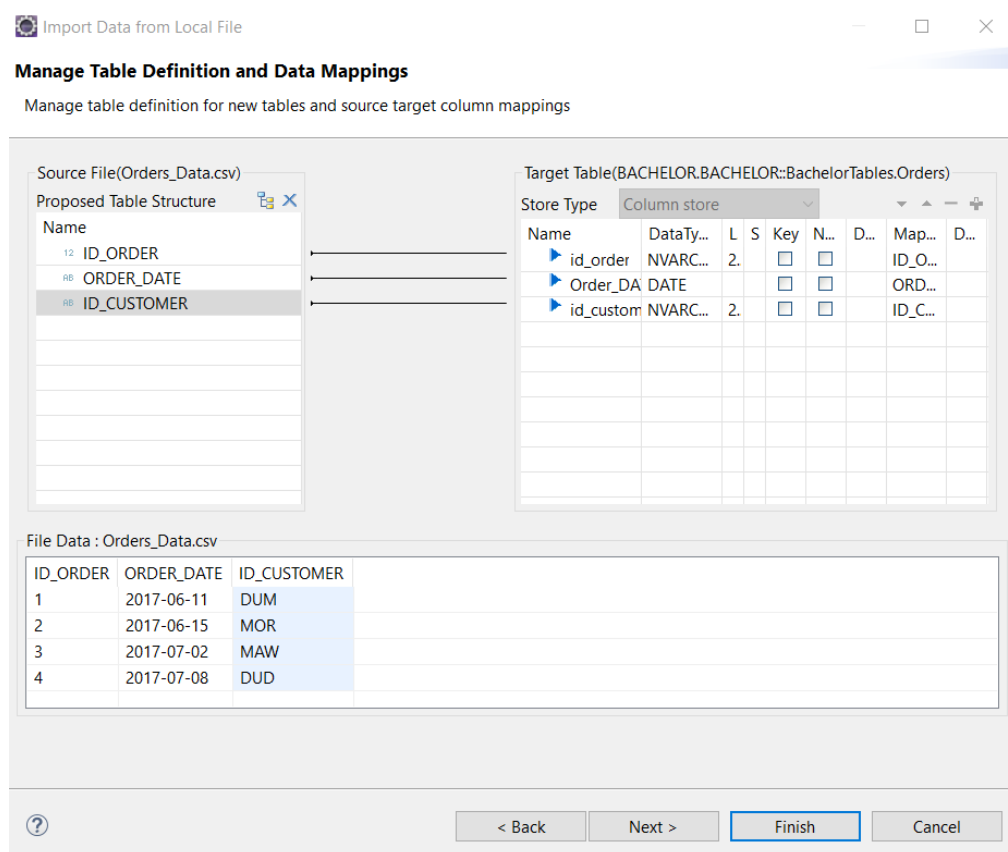


Figure 87 : Annexe IV - Import Data from Local File - Mappings⁸⁷

⁸⁷ Source : Donnée de l'auteur sur Eclipse Neon

Le dernier écran résume les données importées et les éventuelles données déjà existantes dans la table.

Import Data from Local File

Summary
Click Finish to confirm

Details
Source file name: C:\Users\ducma\Desktop\HES_6emeSemestre\Travail Bachelor\Orders_Data.csv
Target Table Name: BACHELOR::BachelorTables.Orders
Target Table Schema: BACHELOR

Data from File

id_order	Order_DATE	id_customer.id...
1	2017-06-11	DUM
2	2017-06-15	MOR
3	2017-07-02	MAW
4	2017-07-08	DUD

Data from Existing Table

id_order	Order_DATE	id_customer.id...

< Back Next > **Finish** Cancel

Figure 88 : Annexe IV - Import Data from Local File - Summary⁸⁸

Il est alors possible d'afficher les données de la table en utilisant *Open Data Preview* et de vérifier la bonne insertion des données.

"BACHELOR". "BACHELOR::BachelorTables.Orders"

Raw Data Distinct values Analysis

Filter patte... 4 rows retrieved - 19 ms

id_order	Order_DATE	id_customer.id_customer
1	11 juin 2017	DUM
2	15 juin 2017	MOR
3	2 juil. 2017	MAW
4	8 juil. 2017	DUD

Figure 89 : Annexe IV - Résultat⁸⁹

⁸⁸ Source : Donnée de l'auteur sur Eclipse Neon

⁸⁹ Source : Donnée de l'auteur sur Eclipse Neon

Annexe V : Gestion de projet

Tout au long de la réalisation de ce travail, la méthodologie Agile étudiée durant le cursus académique a été adaptée au fait que le projet a été réalisé par une seule personne.

Seuls certains points de cette méthodologie ont été retenus et utilisés de manière adaptée aux besoins. En début de travail, un cahier des charges a été mis en place, relevant les objectifs visés. L'utilisation de *Sprints* n'a pas été mise en place mais a été remplacée par des rapports hebdomadaires traitant des objectifs et de leur réalisation, en spécifiant les problèmes rencontrés et les solutions trouvées. Dans la méthodologie Agile, les membres d'une équipe connaissent le domaine et les capacités de l'équipe à effectuer les tâches et sont généralement aptes à évaluer le temps nécessaire à la réalisation de chaque tâche. Devant travailler sur un sujet inconnu, j'ai choisi l'option de ne pas évaluer au départ. En revanche, j'ai tenu un journal hebdomadaire des heures consacrées à chaque tâche. Les rendez-vous avec le Professeur Dr. Werner Maier étaient fixés chaque semaine ou chaque deux semaines en fonction des tâches à effectuer, en vue de faire régulièrement le point sur l'avancement des travaux. En cas de question essentielle entre les différents rendez-vous, il était facile de prendre contact avec lui ou avec Monsieur Frédéric Morand.

Le cahier des charges initial a été modifié pour la partie du prototype HANA CDS. S'il était initialement prévu sur HANA Express (version simplifiée de SAP HANA), il a finalement été réalisé sur SAP HANA, d'une part en raison du manque d'espace sur mon ordinateur (un minimum de 32 GB RAM est requis) et d'autre part en raison de la possibilité d'utiliser un système test existant et disponible auprès du Centre de Compétences SAP ACC Switzerland.

La première semaine de travail s'est focalisée sur la rédaction du cahier des charges, l'établissement d'un premier plan provisoire, ainsi que sur la fixation d'un rendez-vous avec le responsable du travail. Les semaines de travail ont ensuite été réparties de la manière suivante, tenant compte du fait que les cours et les examens se sont terminés en 8^{ème} semaine :

- Environ quatre semaines pour les travaux de recherche et la rédaction de la partie théorique ;
- Environ six semaines pour mettre en place les prototypes et poursuivre les recherches afin de bien comprendre le fonctionnement et l'implémentation du CDS et rédiger la partie pratique ;
- Deux semaines pour la finalisation des prototypes et des documents, ainsi que pour la relecture et les corrections finales.

Annexe VI : Cahier des charges

Cahier des charges

Caractéristiques des tâches à effectuer pour le travail de Bachelor :

- Concept et explication du CDS ;
- Lister et expliquer les différentes fonctionnalités disponibles ;
- Explication de l'ABAP CDS, HANA CDS ;
- Différence entre ABAP CDS et HANA CDS ;
- Avantages et inconvénients par rapport à l'ABAP Dictionary ;
- Création d'un prototype ABAP CDS ;
- Création d'un prototype HANA CDS (~~+ Installation SAP HANA Express~~) ;
- Prise de décision : Est-ce mieux de passer à SAP CDS ou de rester à ABAP Dictionary ?;
- Quelles sont les étapes futures, l'évolution ?

Annexe VII : Détails des heures

Le détail des heures consacrées au travail de Bachelor est exposé ci-dessous. Le CD se trouvant sur la dernière page de ce rapport permet de visualiser à travers les différentes semaines le détail des tâches réalisées et les heures effectuées chaque jour.

</

Figure 90 : Annexe VII - Détails des heures⁹⁰

⁹⁰ Source : Donnée de l'auteur

Annexe VIII : Rapport hebdomadaire

Un exemple de rapport hebdomadaire est illustré ci-dessous. La totalité des rapports est disponible sur le CD se trouvant sur la dernière page de ce rapport.

The figure displays five screenshots of a weekly report form, arranged in a 2x2 grid with a fifth screenshot centered below. Each form is for a student named Marlène Duc, working on a Bachelor Thesis titled 'SAP Core Data Services (CDS)' for Week 01 (01.05.2017 – 07.05.2017). The forms show the progression of the report's content over time.

- Top Left Screenshot:** The 'Initial situation' section contains three bullet points: 'Comprendre le but du travail de bachelor', 'Chercher des liens sur le CDS pour comprendre le contexte', and 'Lecture du guide et mise en page du rapport'. The 'Problem(s)' section is empty.
- Top Right Screenshot:** Identical to the top left, showing the 'Initial situation' section.
- Middle Left Screenshot:** The 'Problem(s)' section is empty.
- Middle Right Screenshot:** The 'Planning for the next week' section contains two bullet points: 'Recherche de liens sur les technologies et les différents prérequis pour l'utilisation du SAP CDS' and 'Rédaction suite aux recherches'.
- Bottom Center Screenshot:** The 'Resolved problem(s)' section is empty.

Figure 91 : Annexe VIII - Rapport hebdomadaire⁹¹

⁹¹ Source : Donnée de l'auteur

Contacts

La rédaction de ce travail a été effectuée en collaboration avec les personnes suivantes :

Professeur responsable

Dr. Werner Maier

Domaine Economie & Services

werner.maier@hevs.ch

Adjoint scientifique

M. Frédéric Morand

Domaine Economie & Services

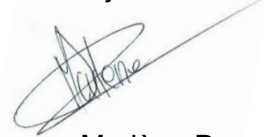
frederic.morand@hevs.ch

Pour l'obtention des informations liées aux systèmes utilisés et les mots de passe, merci de prendre contact avec l'une de ces personnes.

Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seule, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après : Prof. Dr. Werner Maier – M. Frédéric Morand.

Crans-Montana, le 28 juillet 2017



Marlène Duc